

MANUAL DE PRACTICAS INTERFACES GRAFICAS APLICACIONES DE LA IOT ARDUINO Y LABVIEW

ELABORÓ:

JOAQUIN IRIGOYEN BAUTISTA

ABRIL 2021

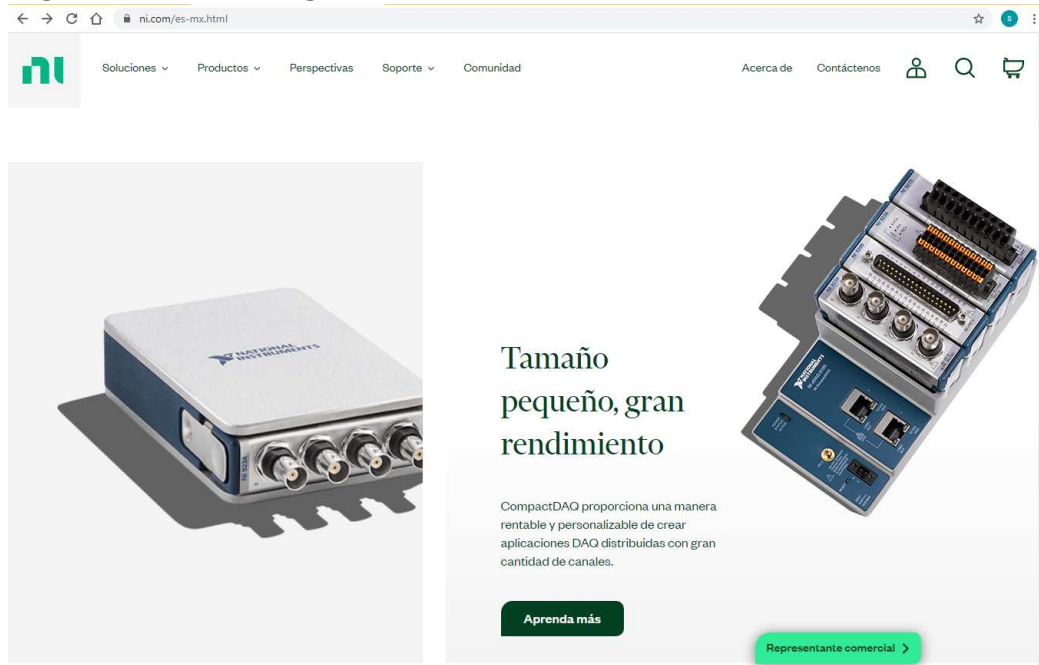
Índice

Tabla de contenido

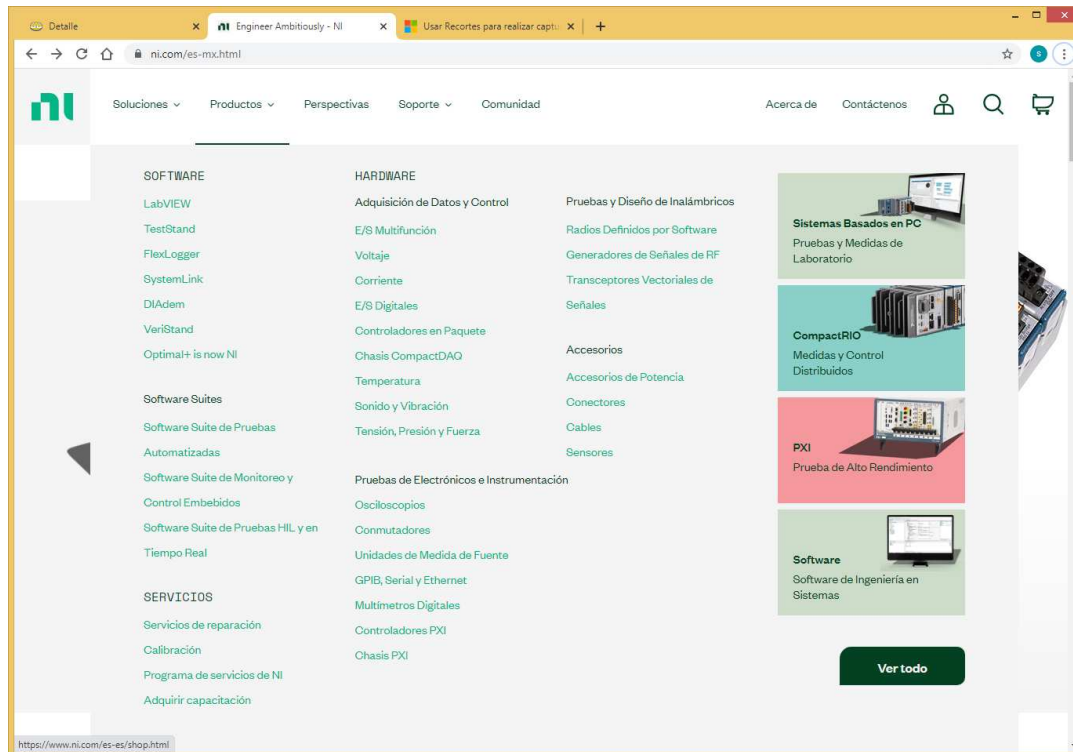
<i>Practica 1: instalación de labview</i>	<i>2</i>
<i>PRACTICA 2: Conociendo Labview despues de instalarlo</i>	<i>11</i>
<i>PRACTICA 3 ENTRADAS SALIDAS EN LABVIEW.....</i>	<i>29</i>
<i>PRACTICA 4: ENTRADAS Y SALIDAS DIGITALES O BOOLEANAS.....</i>	<i>33</i>
<i>PRACTICA 5: Suma de dos numeros usando controles numericos con LabView</i>	<i>37</i>
<i>PRACTICA 6: Comparación entre 2 numeros y se muestra si son iguales o diferentes..</i>	<i>42</i>
<i>PRACTICA 7 : TIPOS DE DATOS</i>	<i>48</i>
<i>PRACTICA 8: SUMA DE DOS NUMEROS USANDO VARIABLES.....</i>	<i>60</i>
<i>PRACTICA 9: OPERACIONES MATEMATICAS BASICAS ENTRE 2 NUMEROS</i>	<i>66</i>
<i>PRACTICA 10: APLICACIÓN USANDO ESTRUCTURA CASE</i>	<i>70</i>
<i>Practica 11 Arduino con LabView.....</i>	<i>79</i>
<i>PRACTICA 12: Lectura de un boton desde Arduino, monitoreado por Labview.....</i>	<i>106</i>
<i>PRACTICA 13: Leer un dato analogico desde Arduino y mostrar en un indicador tipo aguja en LabView.....</i>	<i>127</i>

Practica 1: instalación de labview

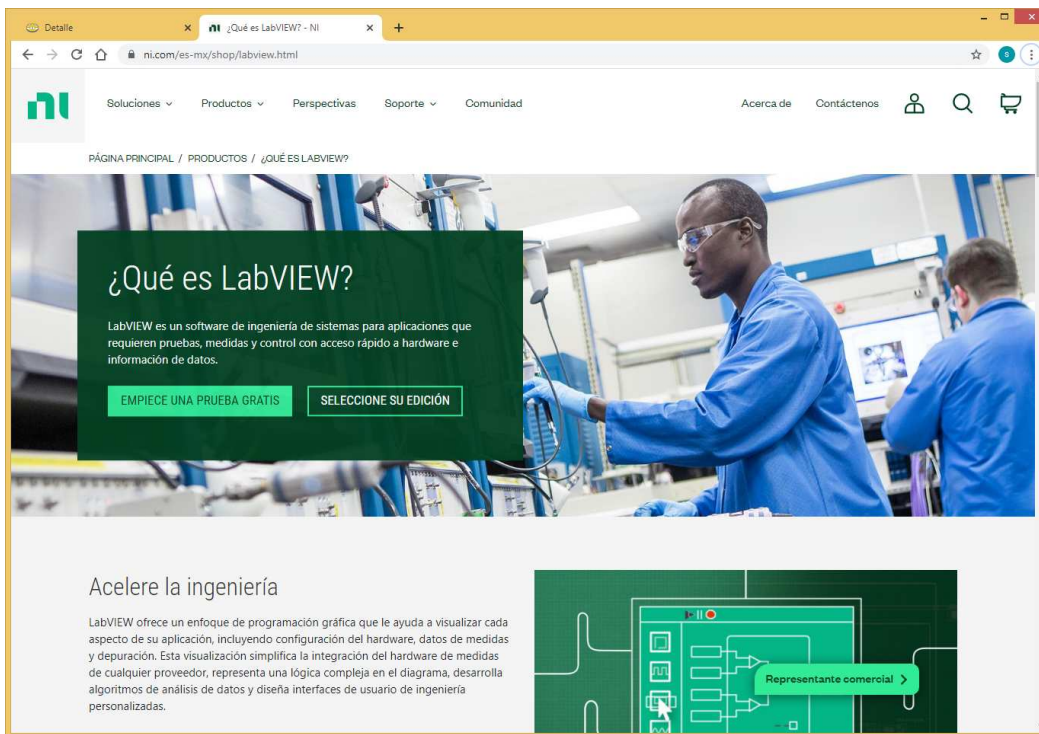
Ingresamos a la página: <https://www.ni.com/es-mx.html>



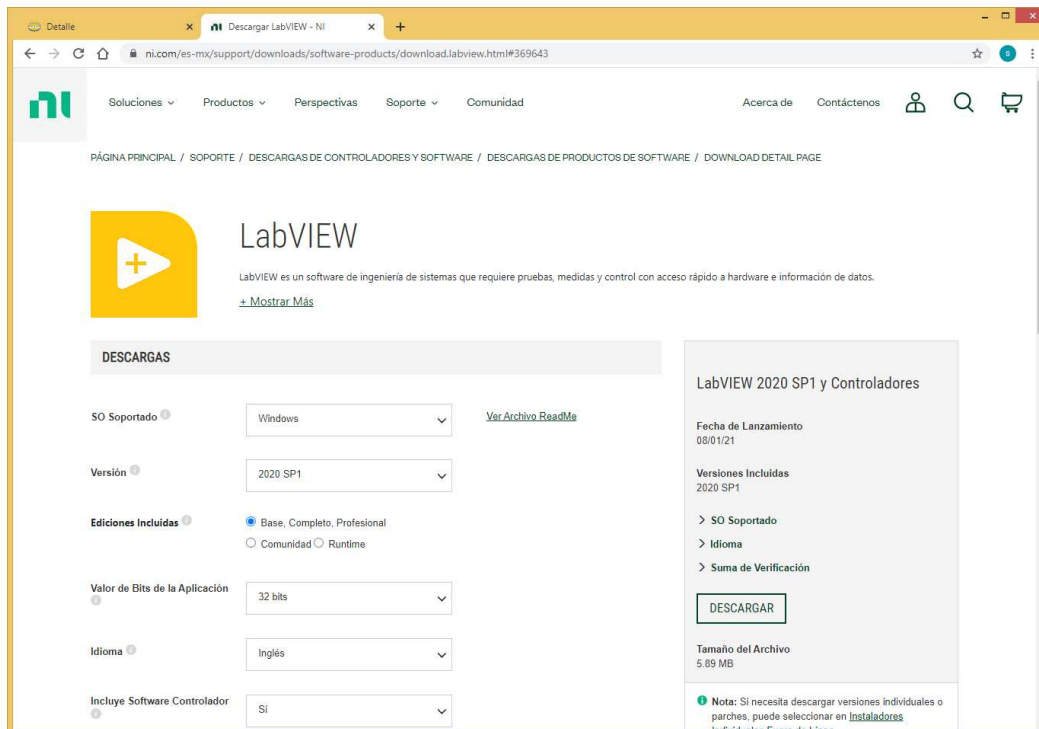
Clic en el menu Productos. Elegimos Software y nos mostrará la opción: Labview



En esta pantalla, vamos a descargar el instalador online, para ello, clic en “EMPIECE UNA PRUEBA GRATIS”



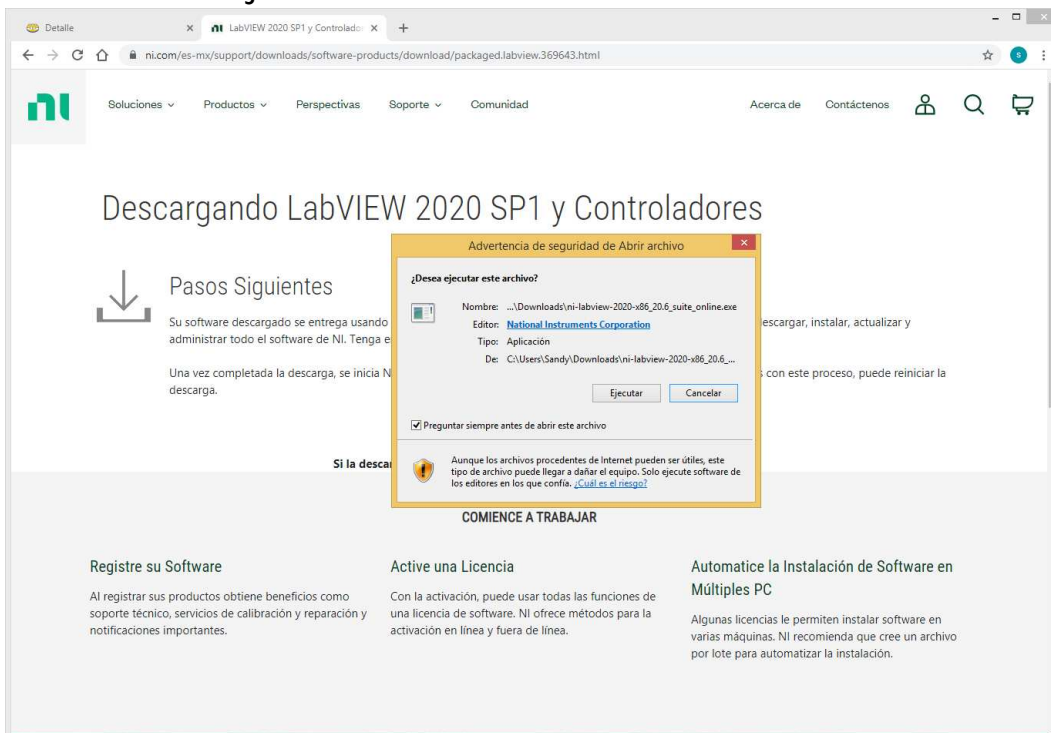
Aparecerán las opciones de el instalador, debemos seleccionar el Sistema Operativo adecuado. En caso de usar Windows, elegimos la versión más reciente (2020 SP1) y que sea de 32 bits, pues así lo recomienda National Instruments.



Clic en Descargar y está comenzara automaticamente.



Se descarga el instalador como se muestra en la figura y posteriormente damos clic en Ejecutar



Se abra el asistente de instalación en línea donde deberemos aceptar los contratos de licencia

NI Package Manager

Select Agree Review Finish

You must accept the license agreements below to proceed.

NI

CONTRATO DE LICENCIA DE SOFTWARE DE NATIONAL INSTRUMENTS

LEA ATENTAMENTE ESTE ACUERDO DE LICENCIA DE SOFTWARE ("ACUERDO"). AL DESCARGAR EL SOFTWARE Y / O HACER CLIC EN EL BOTÓN CORRESPONDIENTE PARA COMPLETAR EL PROCESO DE INSTALACIÓN, SE COMPROMETE A RESPETAR LOS TÉRMINOS DE ESTE ACUERDO. SI USTED NO DESEA SER PARTE DE ESTE ACUERDO Y ESTAR SUJETO A SUS TÉRMINOS Y CONDICIONES, NO INSTALE EL SOFTWARE Y DEVUELVA EL SOFTWARE (CON TODOS LOS MATERIALES QUE LE ACOMPAÑAN Y SUS EMPAQUES) EN UN PLAZO DE TREINTA (30) DÍAS DESDE QUE LO RECIBE. TODAS LAS DEVOLUCIONES A NI ESTARÁN SUJETAS A LAS POLÍTICA DE DEVOLUCIÓN VIGENTES DE NI EN ESE MOMENTO. SI USTED ACEPTA ESTOS TÉRMINOS EN NOMBRE DE UNA ENTIDAD, USTED ACEPTA QUE TIENE AUTORIDAD PARA OBLIGAR A LA ENTIDAD A ESTOS TÉRMINOS.

This license agreement applies to the following packages: NI Package Manager

☐ I do not accept the license agreement. ☒ I accept the above license agreement.

Next

Aquí nos aparecerá la pantalla con el resumen de la instalación

NI Package Manager

Select Agree Review Finish

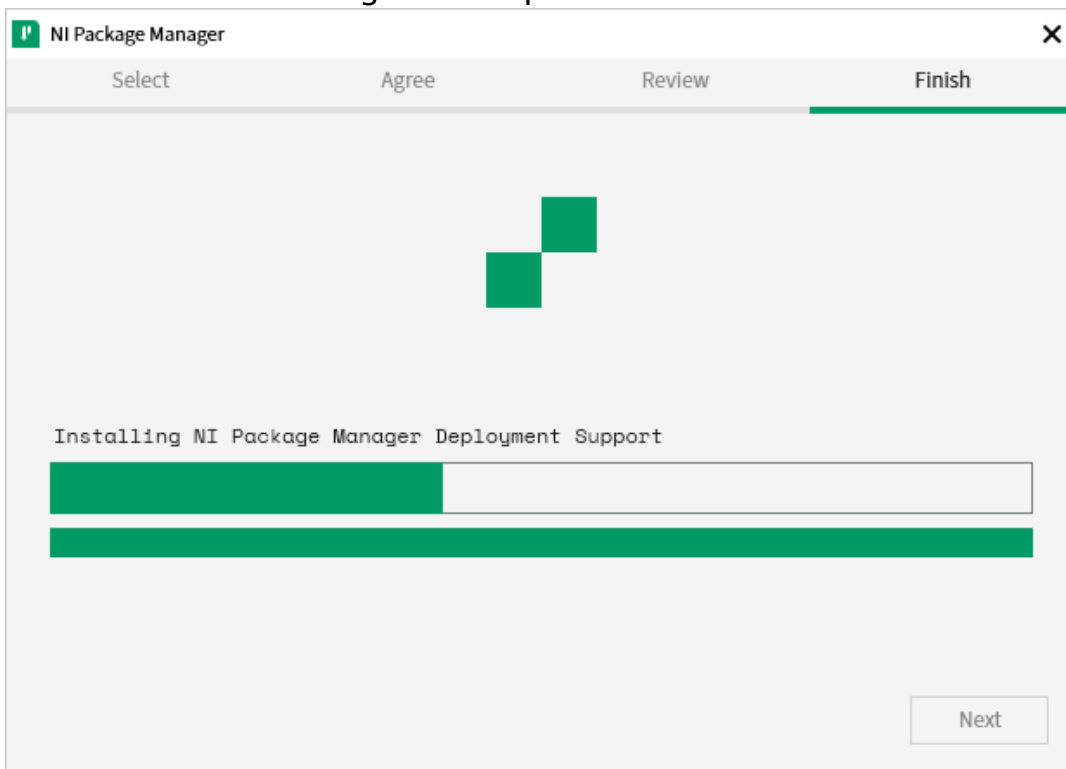
Review the following summary before continuing.

▼ Install

NI Package Manager	20.6.0
--------------------	--------

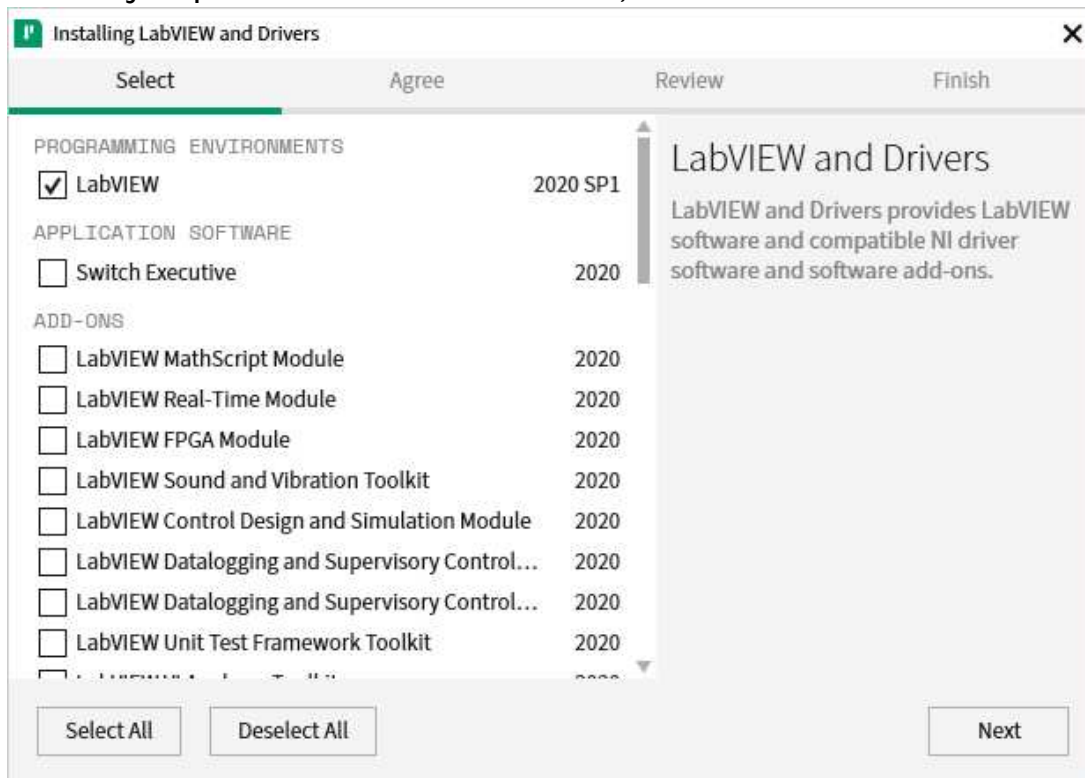
Back Next

Se comienza la descarga de la aplicación básica como muestra la imagen.

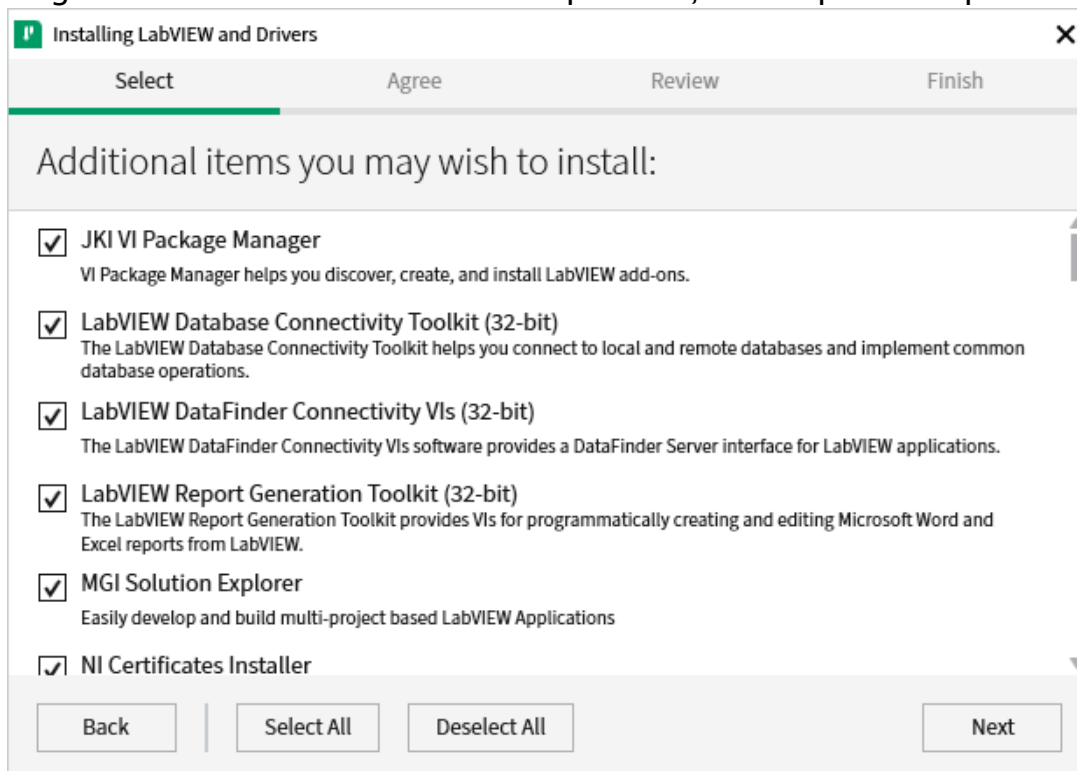


Posteriormente, nos aparecen opciones para la instalacion de Drivers que necesitemos (ADD-ONS)

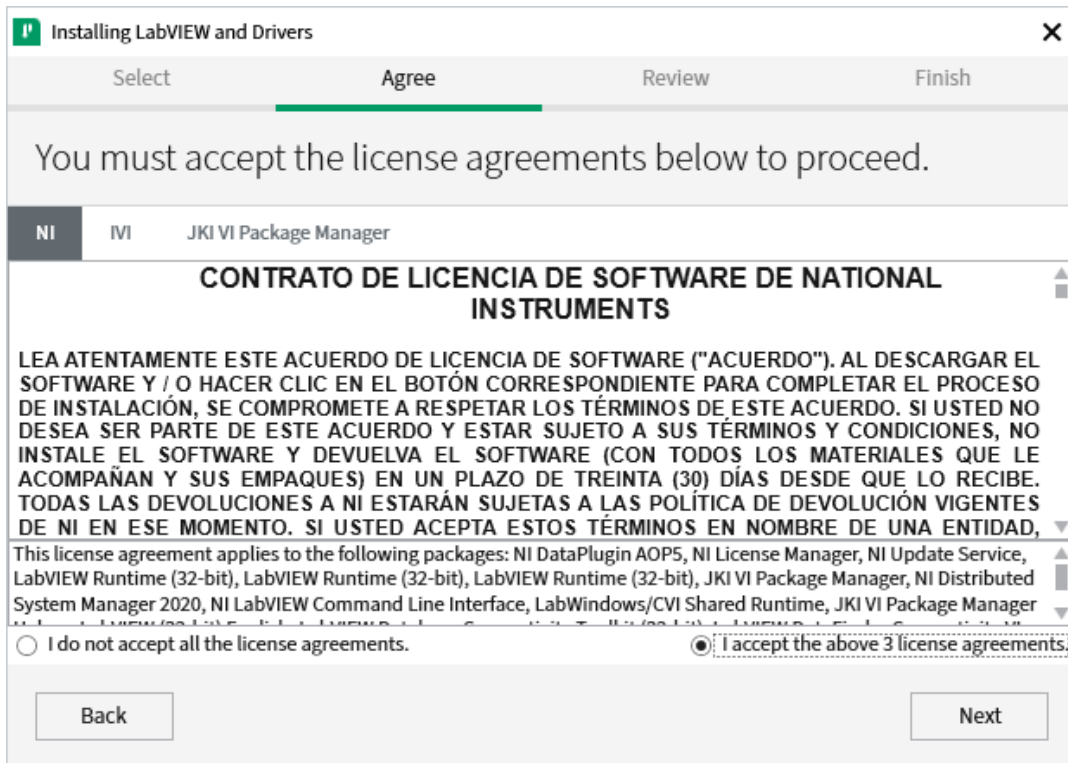
En este caso podríamos dejar solo la opción de Labview, pero como vamos a trabajar puerto serie con Ardiono, necesitamos las librerías VISA



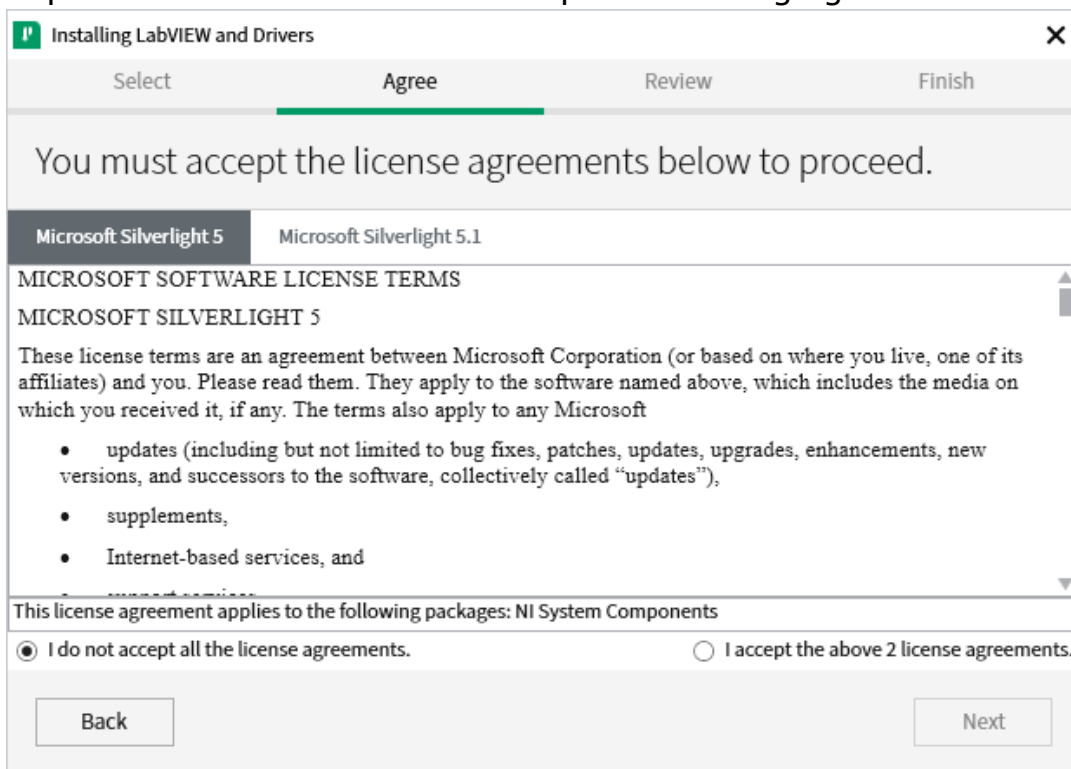
Elegimos como se muestra en pantalla, las opciones por default



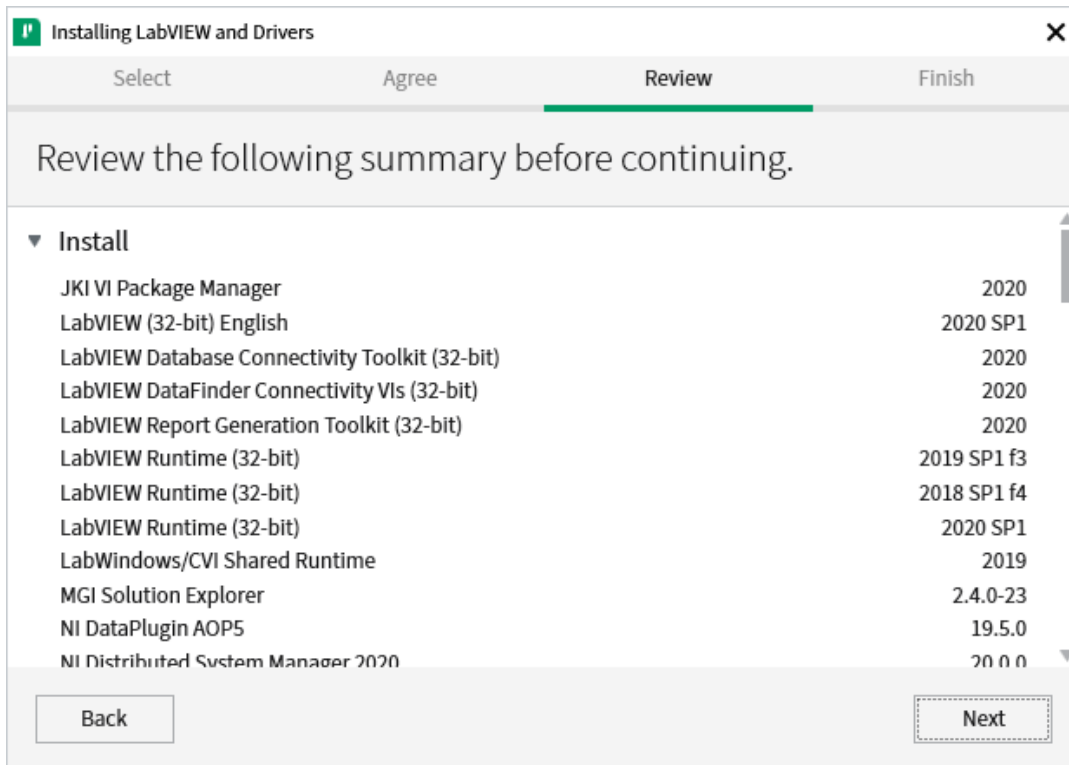
Damos clic en Next y viene un Contrato nuevamente



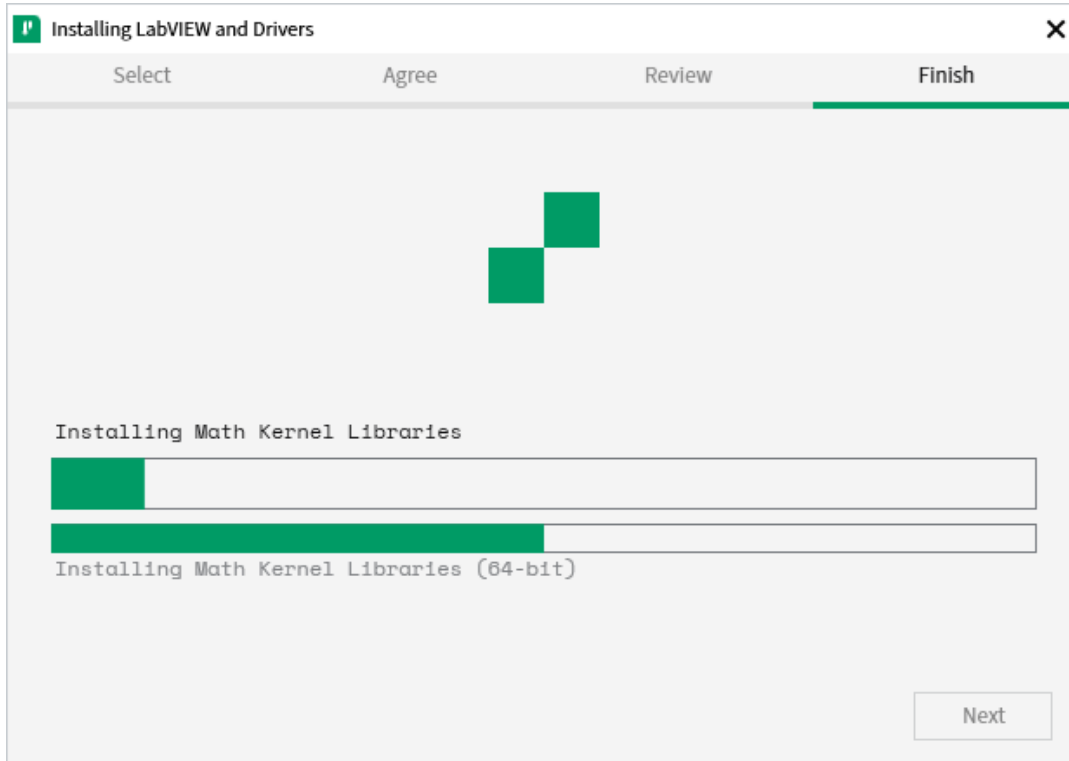
Damos clic Aceptando ese y los demas contratos que nos presente que depende de la cantidad de Drivers que vamos a agregar



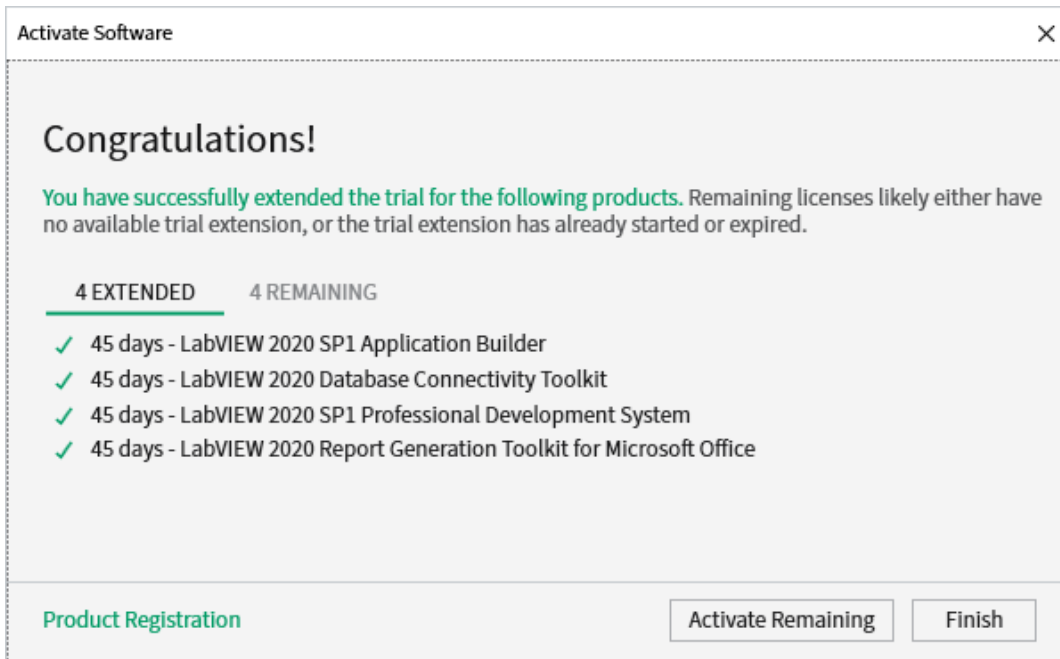
Al final aparecera el resumen de lo que instalara adicionalmente, damos clic en Next



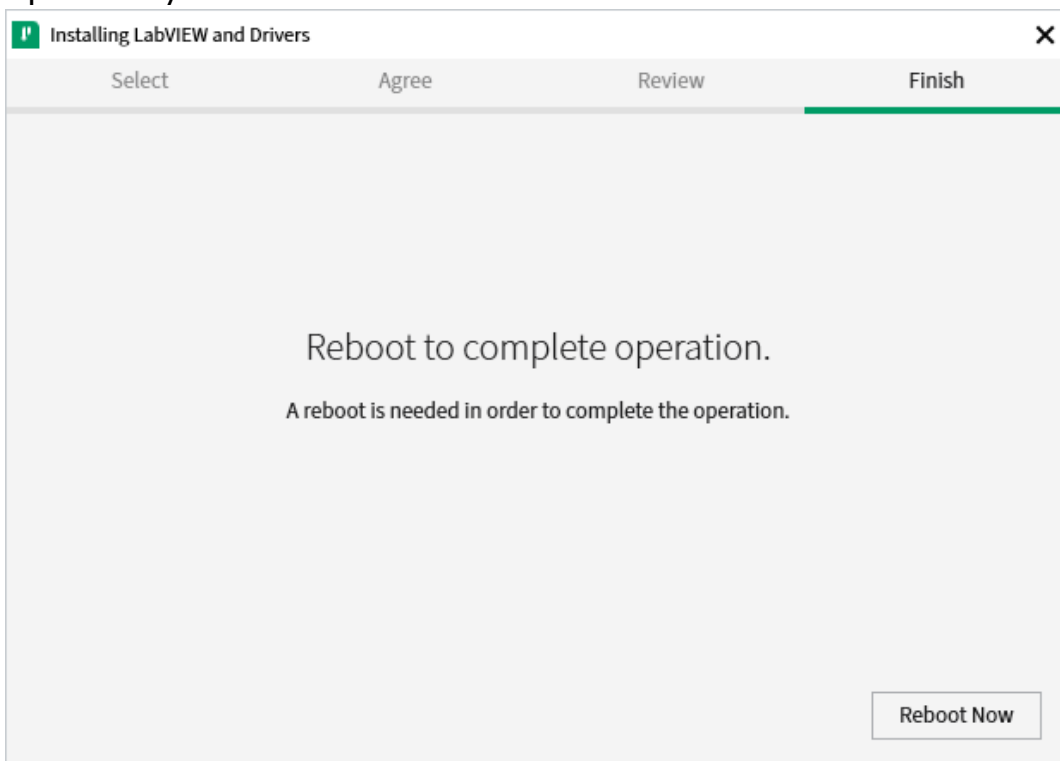
Este proceso puede demorar dependiendo de tu velocidad de internet



Una vez terminada la instalación nos aparece la siguiente ventana, indicándonos que tenemos 45 días de prueba en los elementos que hemos instalado

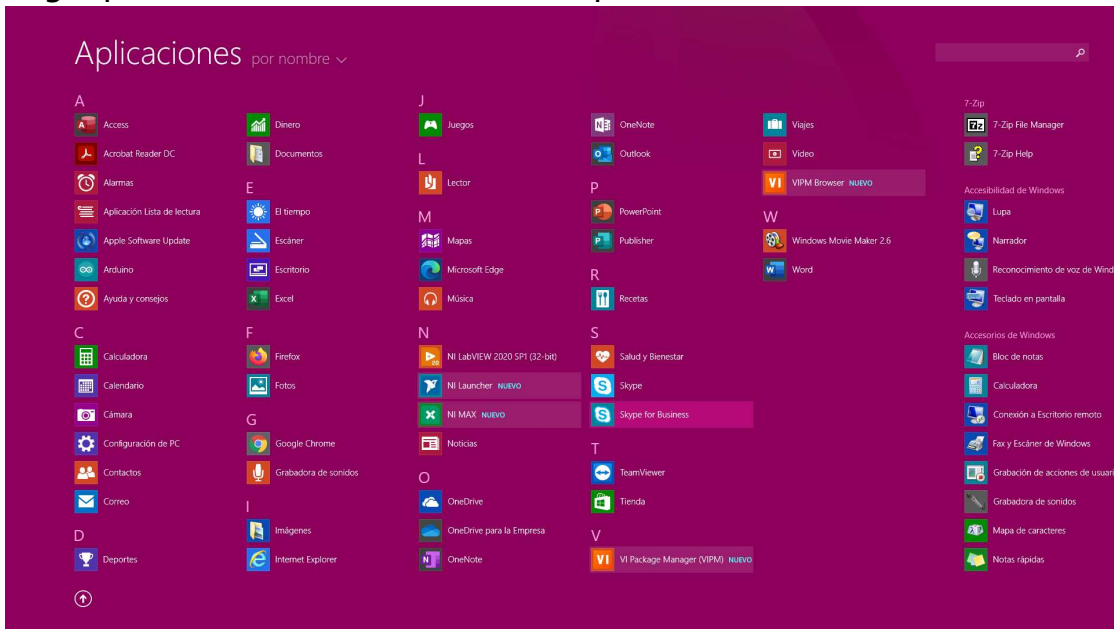


Para comenzar a usar la aplicación, requerimos reiniciar nuestro sistema operativo y listo.

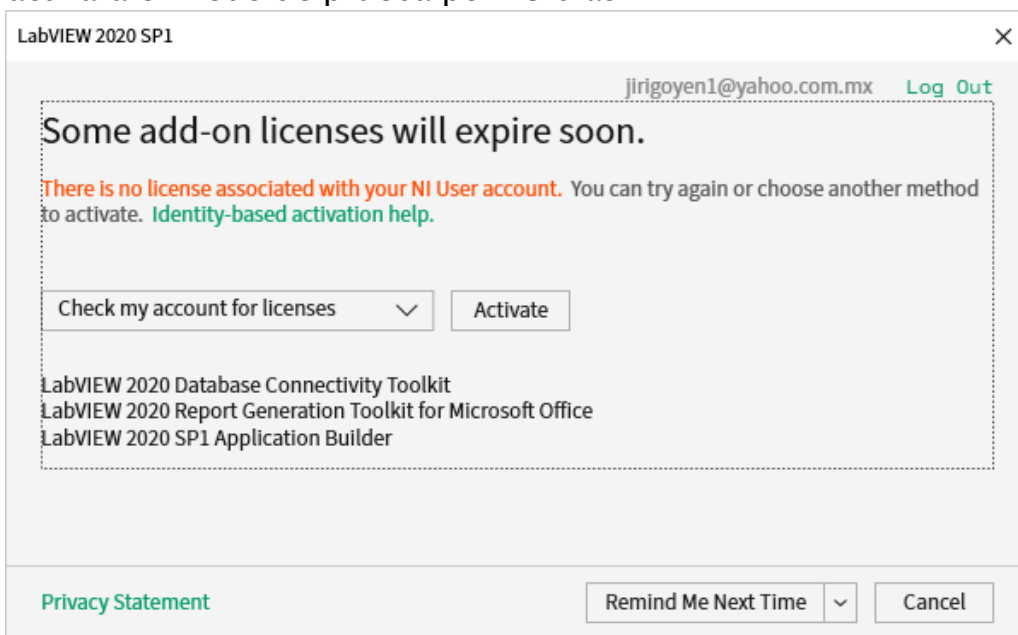


PRACTICA 2: Conociendo Labview despues de instalarlo

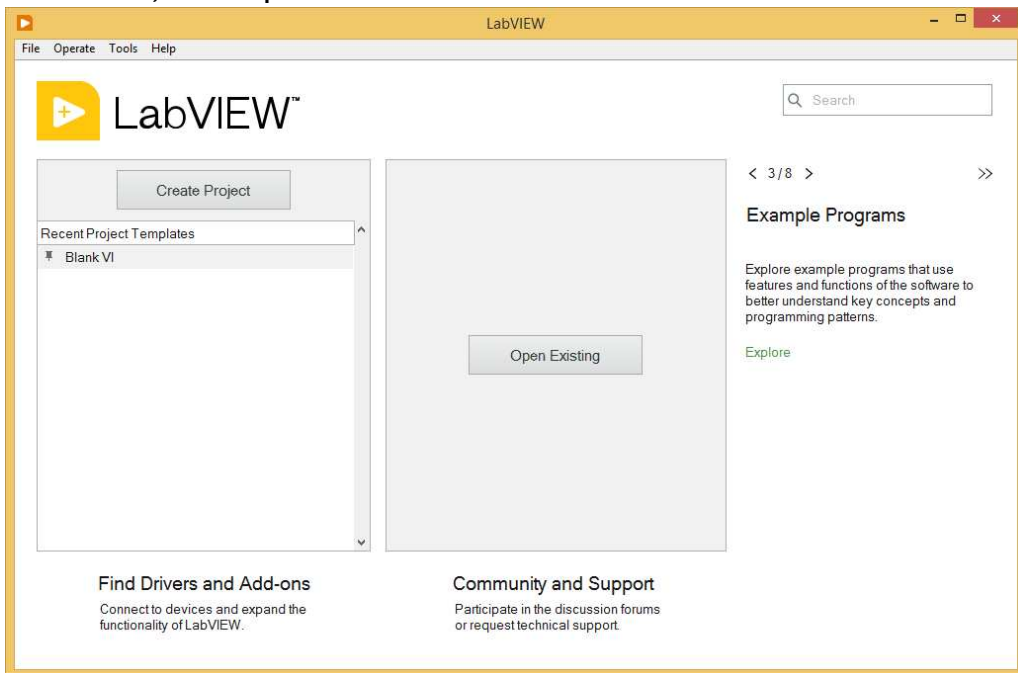
En nuestros programas aparecera NI LabVIEW 2020 SP1 o la versión que se eligio para instalar. Damos doble clic para abrirla



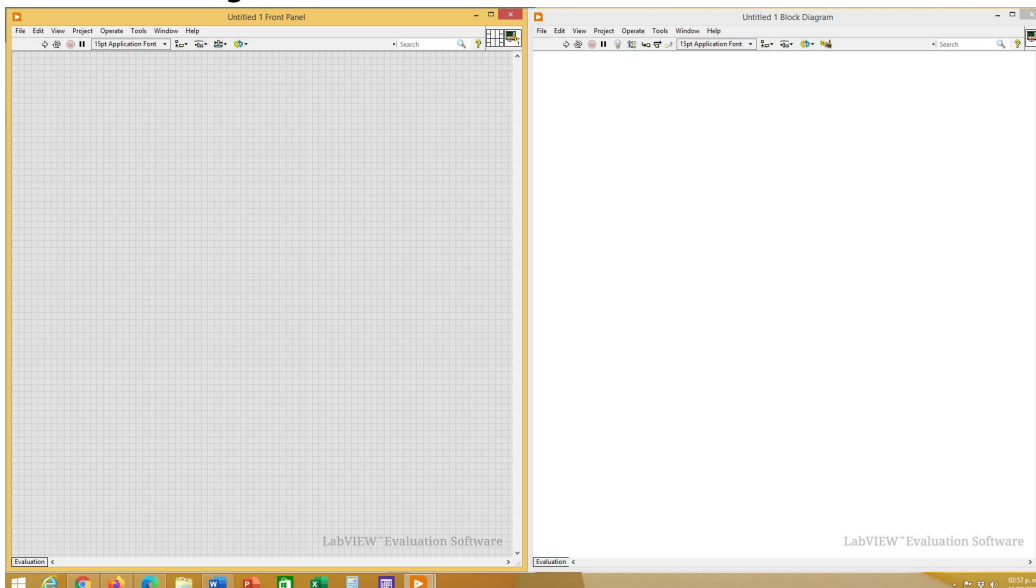
Como no tenemos licencia, debemos comprar una o instalar una licencia de prueba por 45 dias, para esto unicamente damos clic en Check my account for licences y Activate. Al no encontrar una licencia retail, nos activara el modo de prueba por 45 dias.



Abriendo entonces el programa y aceptando la licencia de prueba, encontraremos esta ventana. Aquí con las practicas que se van a realizar, eligiremos Blank VI. Si queremos abrir un programa que ya hemos realizado, este aparecera de lado derecho.

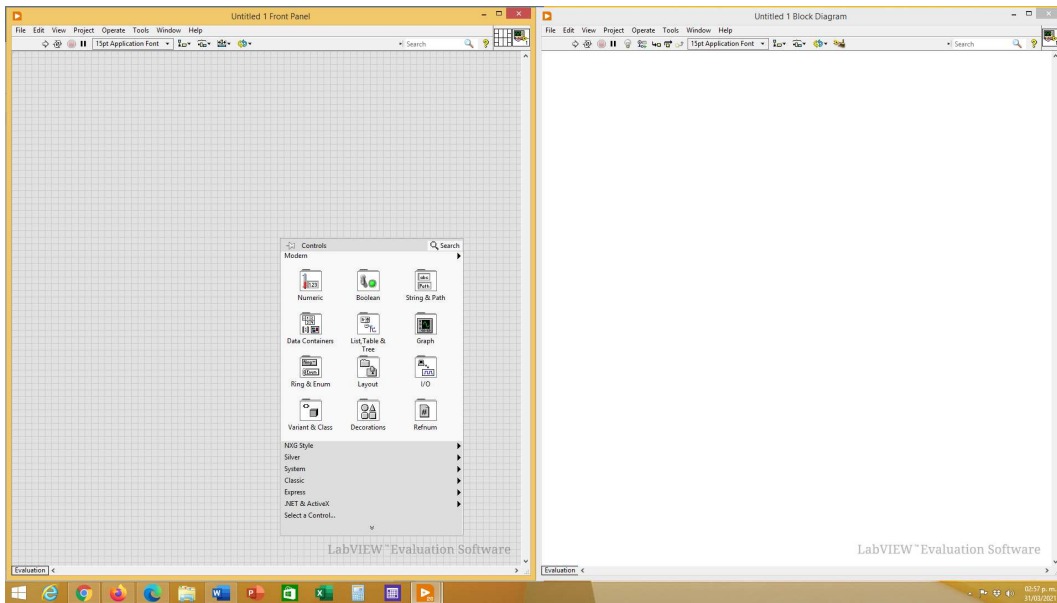


Siguiente paso es que se abran 2 ventanas: Panel Frontal y Bloques. Para que aparezcan juntas una al lado de la otra, en el menu Window, elegimos Tile left and right

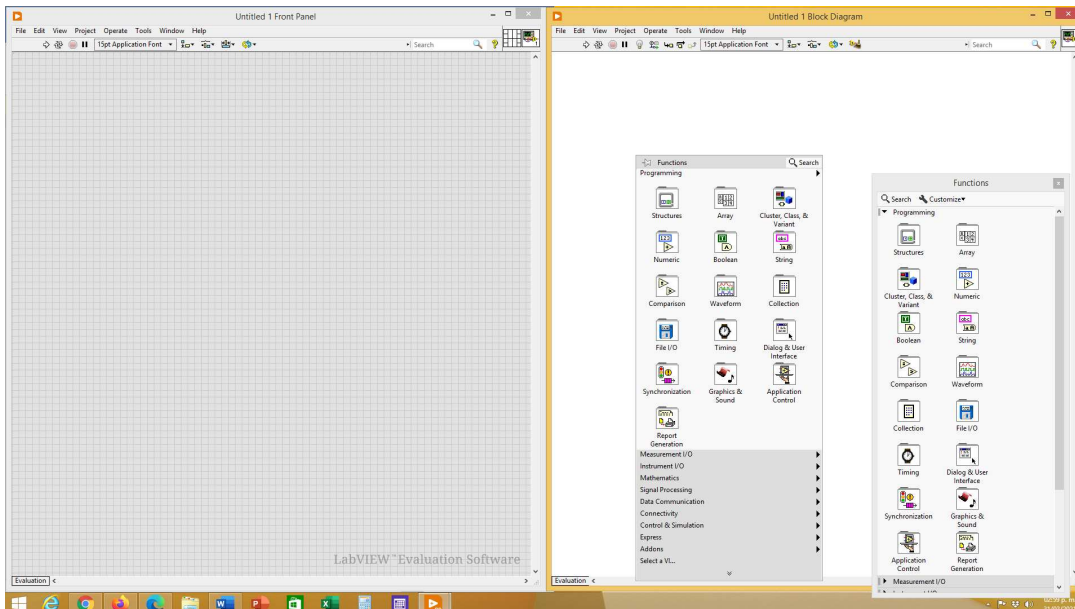


Para mostrar los controles y los bloques, basta con dar clic con boton derecho y apareceran menus de controles que seran diferentes en cada

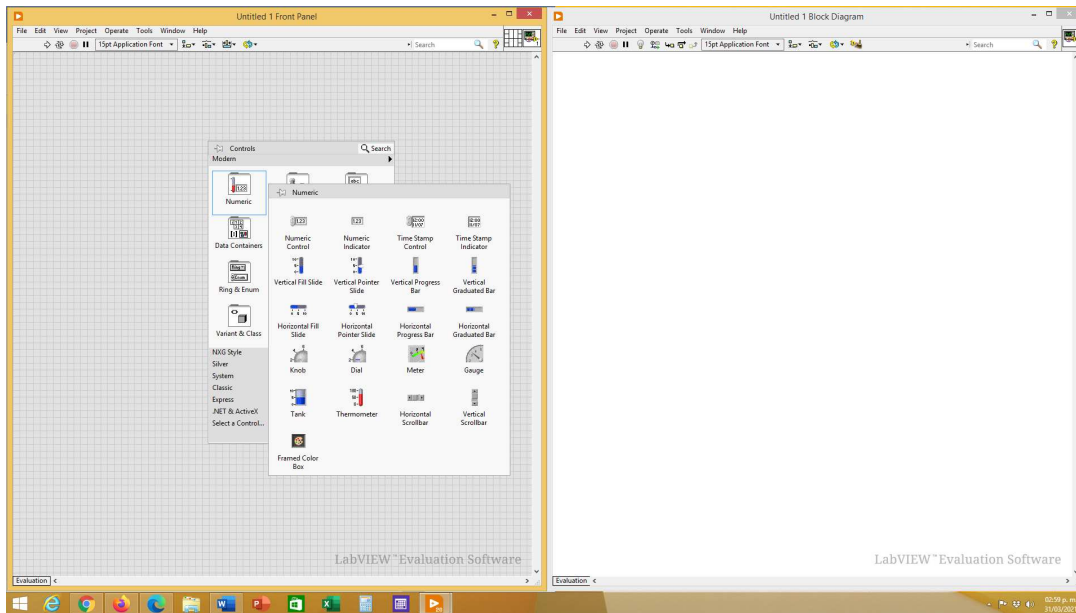
una de las 2 ventanas.



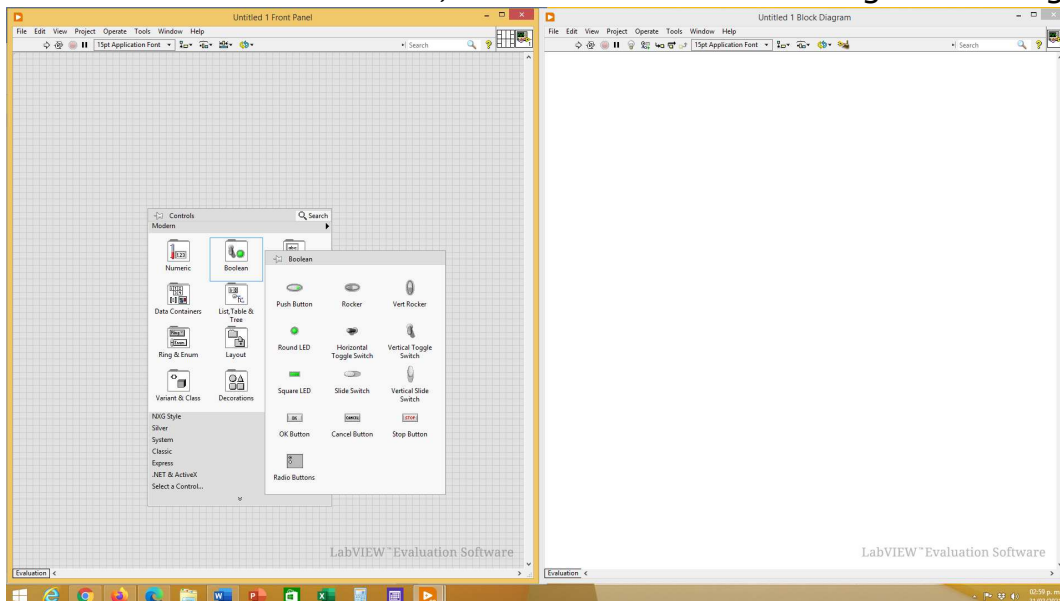
Observamos la diferencia con las funciones de la ventana de Bloques



Veamos cada menu de opciones. El primero es el de controles Numericos, donde veremos tando indicadores como controles, estos se muestran en la siguiente imagen



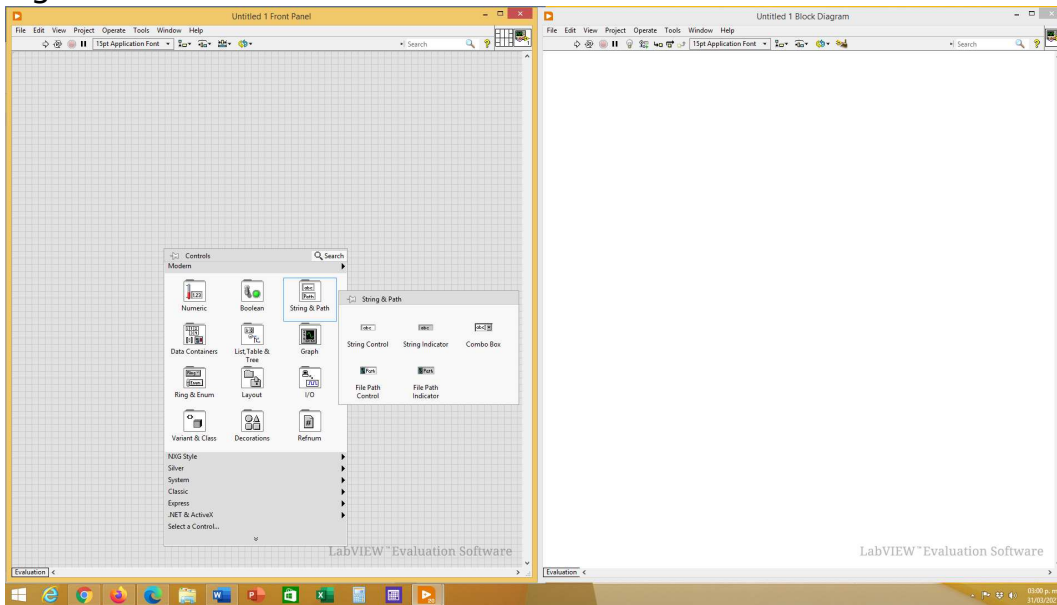
El siguiente es el de controles Booleanos, donde veremos tanto indicadores como controles, estos se muestran en la siguiente imagen



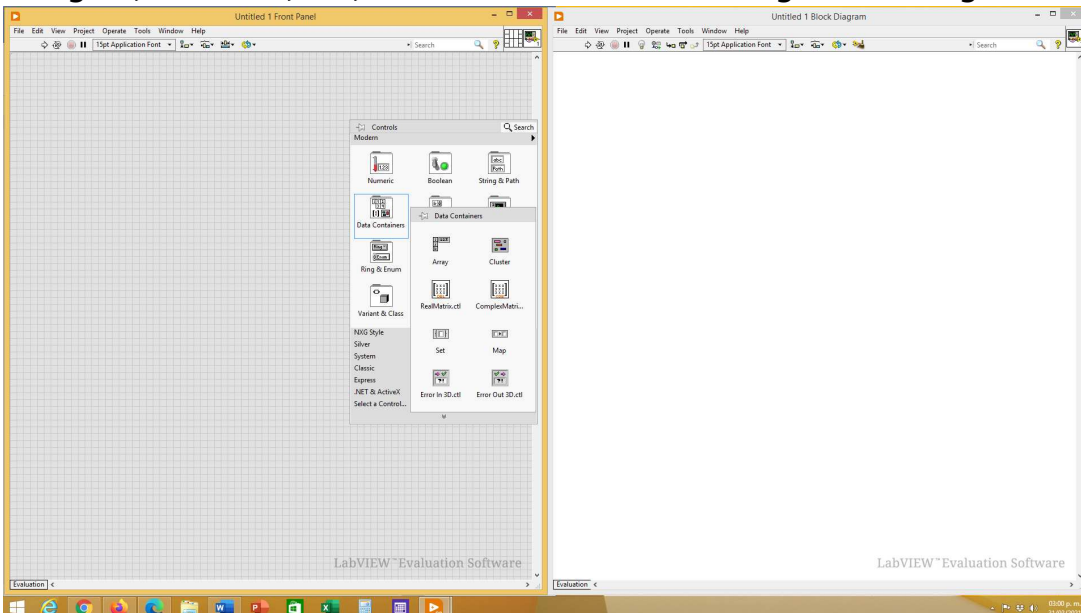
El siguiente es el de controles String, donde veremos tanto indicadores como controles todos relacionados con cadenas, estos se muestran en la

siguiente

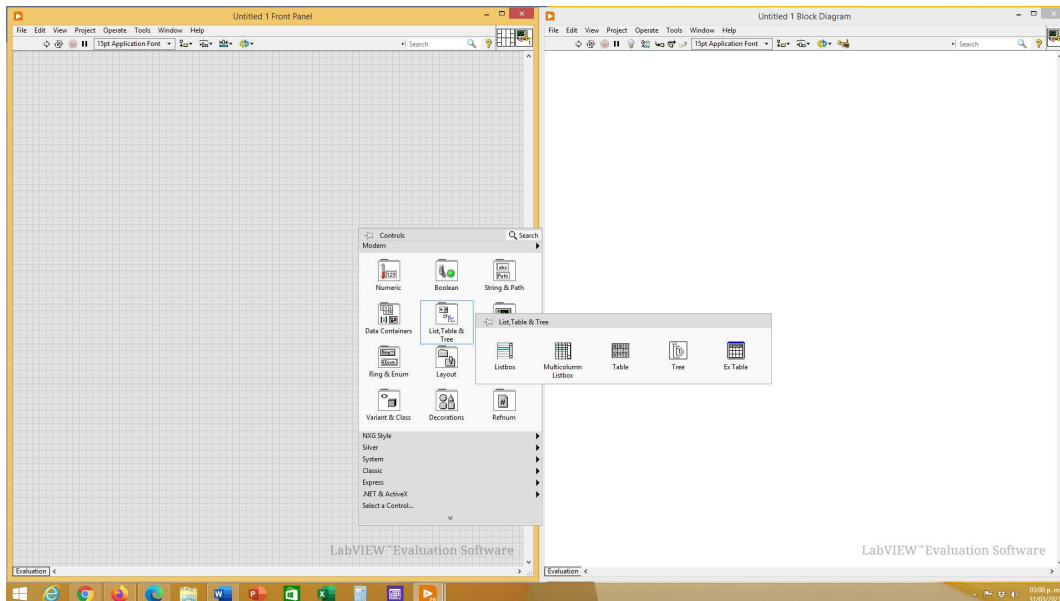
imagen



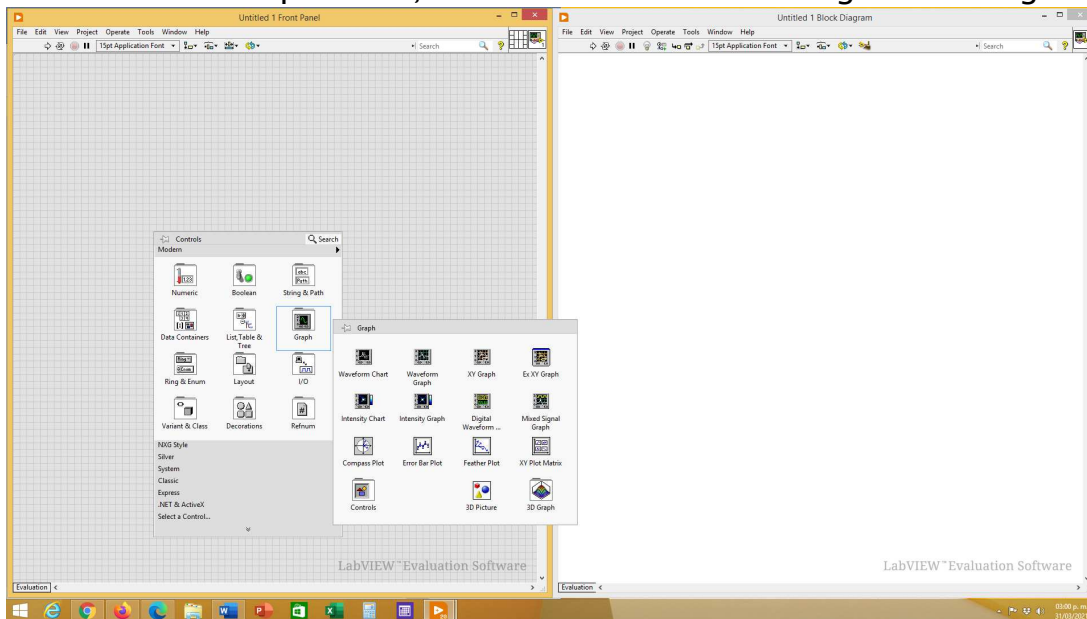
El siguiente es el de controles Contenedores de Datos, donde veremos arreglos, matrices, etc, estos se muestran en la siguiente imagen



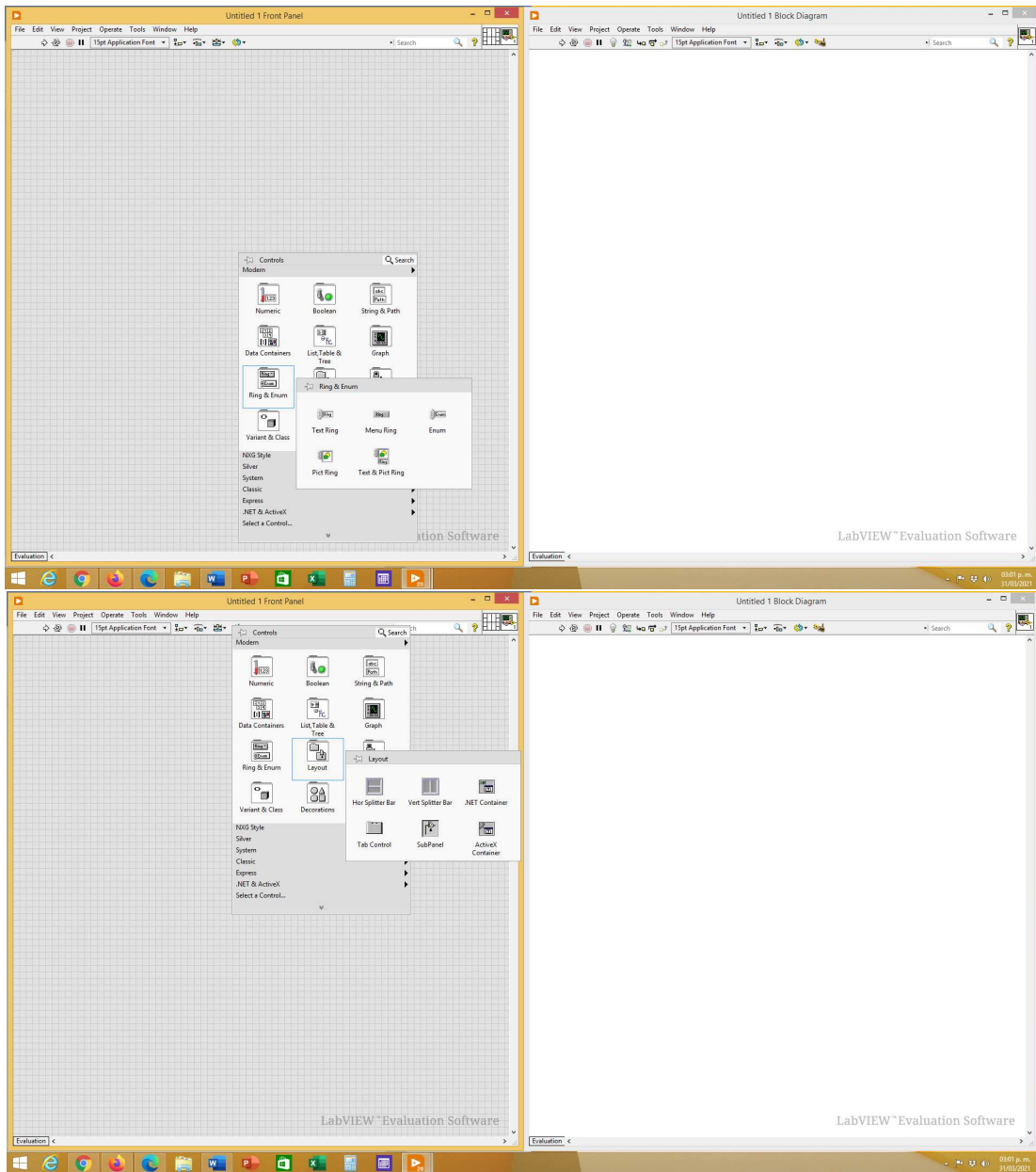
El siguiente es el de controles de Listas, Tablas y Arboles, estos se muestran en la siguiente imagen

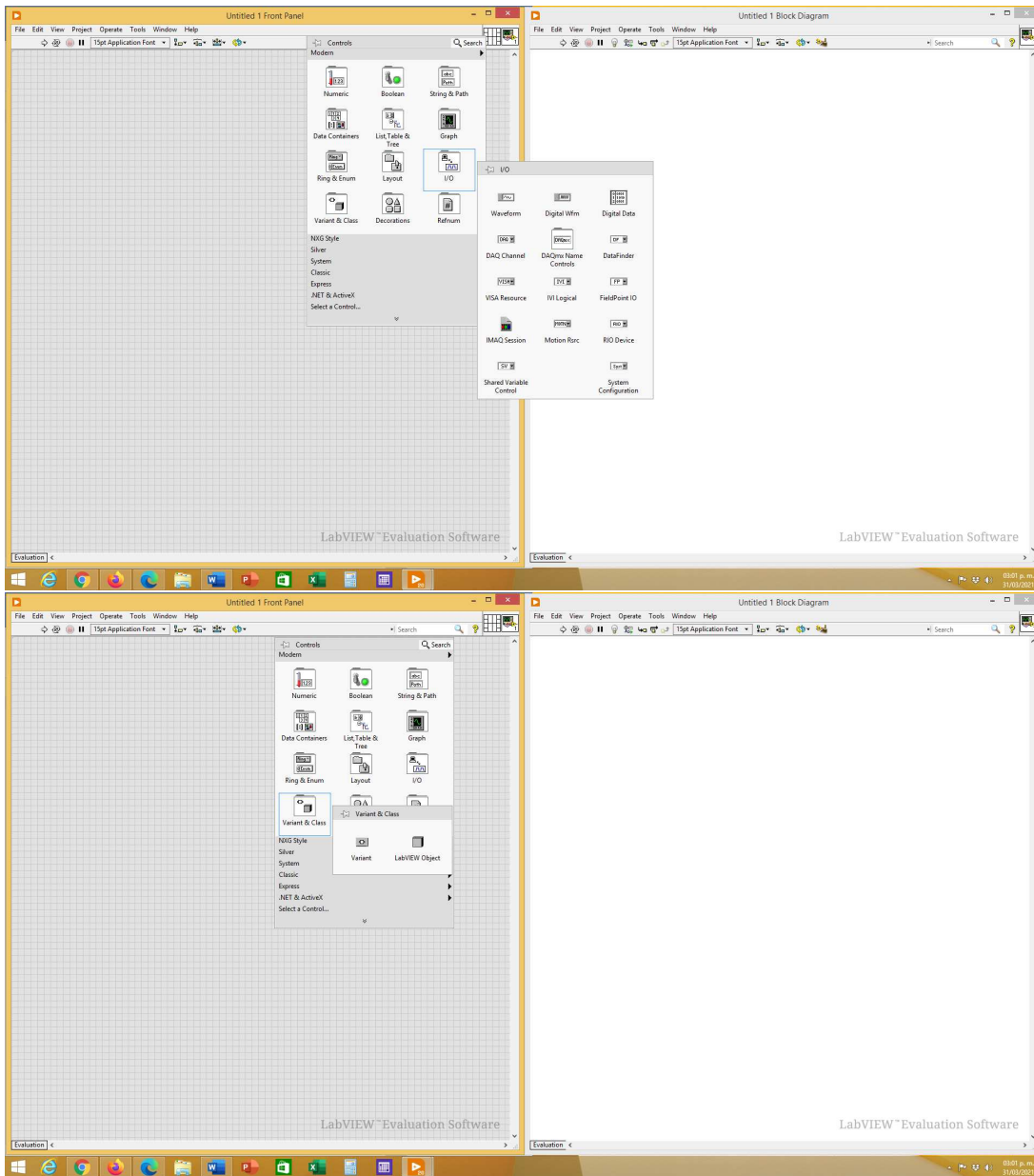


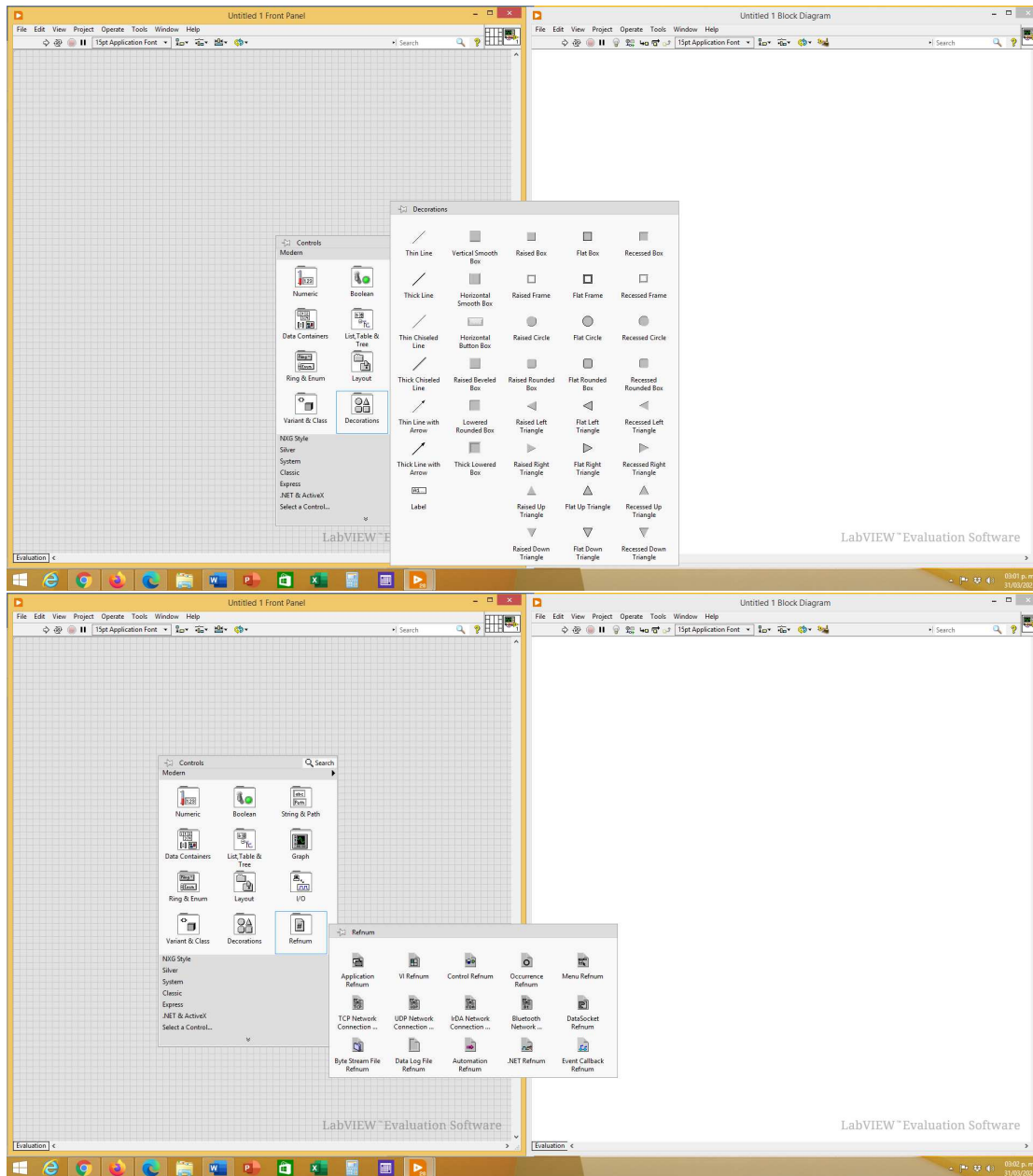
El siguiente es el de controles Graficos, donde veremos tando indicadores como controles tipo Onda, estos se muestran en la siguiente imagen



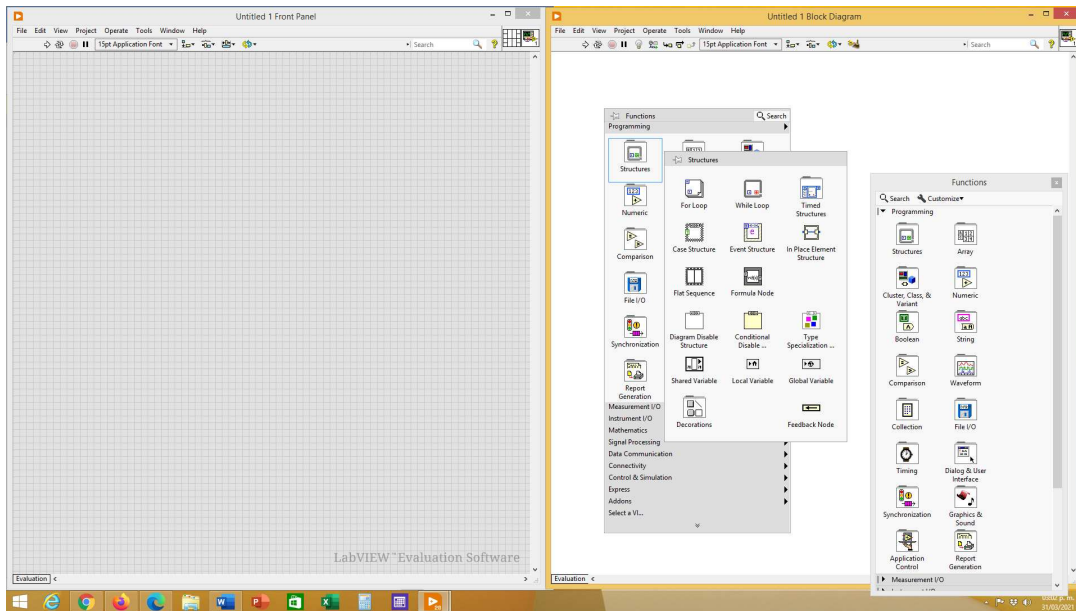
En las siguientes imágenes se muestran los demás menus del panel Frontal como referencia.



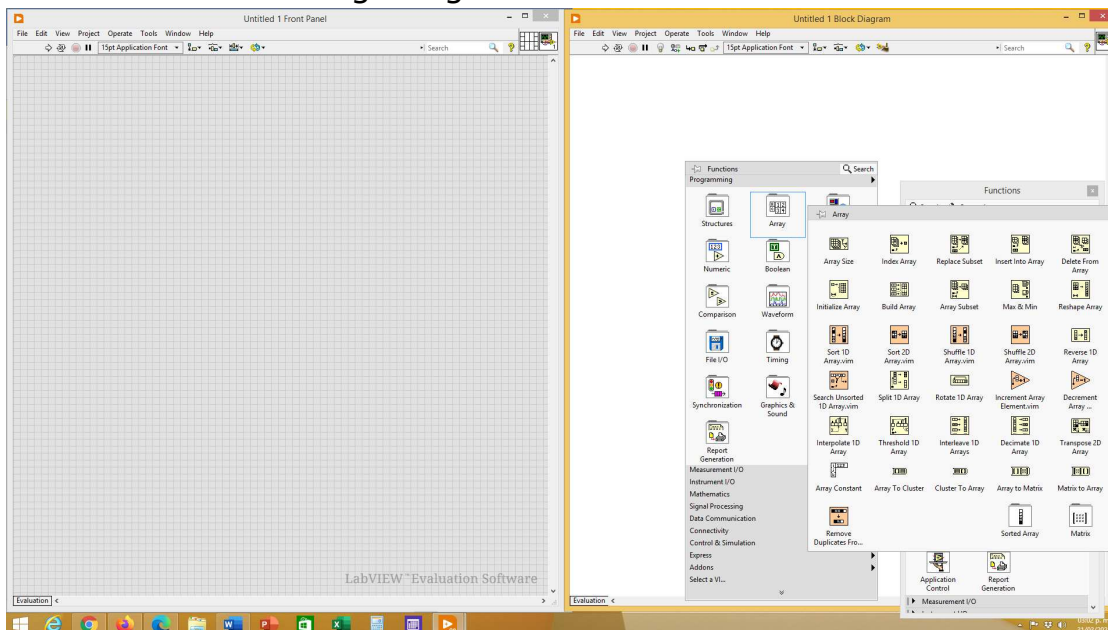




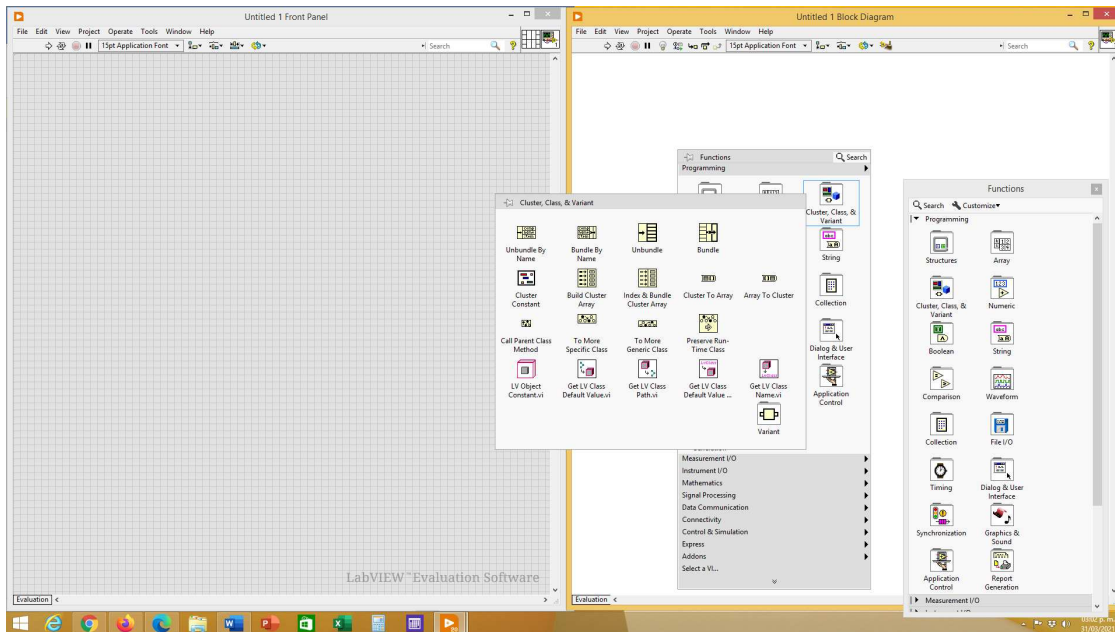
En la ventana de bloques tenemos las Estructuras como primer bloque, aquí encontraremos For Loop, While Loop, Case Structure entre otras



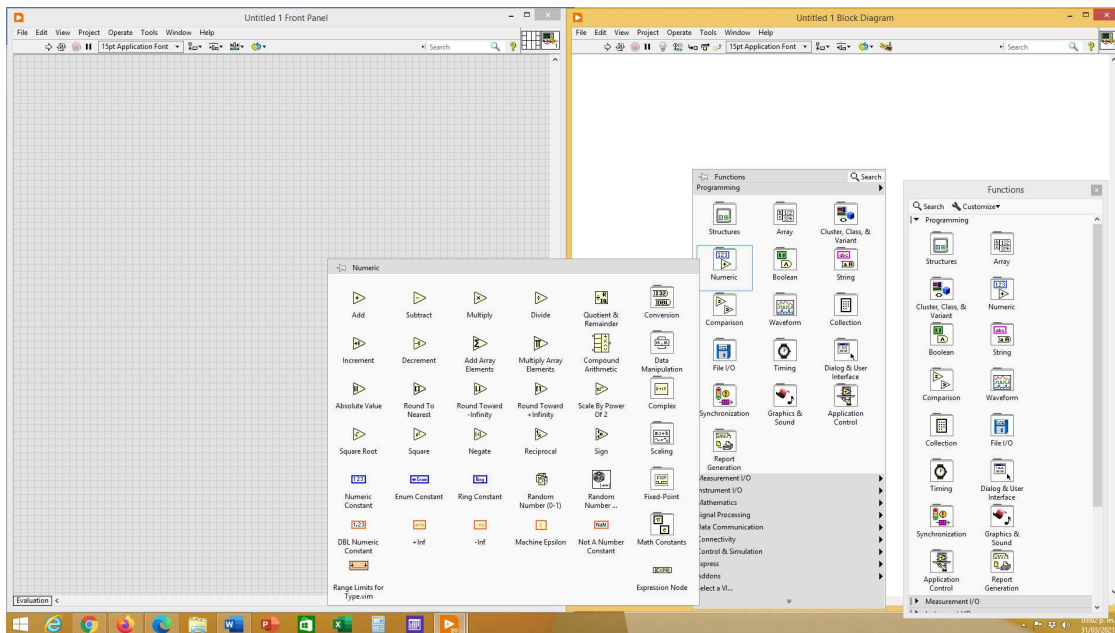
Tenemos otras opciones en el menu de bloques Arreglos, dichas opciones se muestran en la imagen siguiente



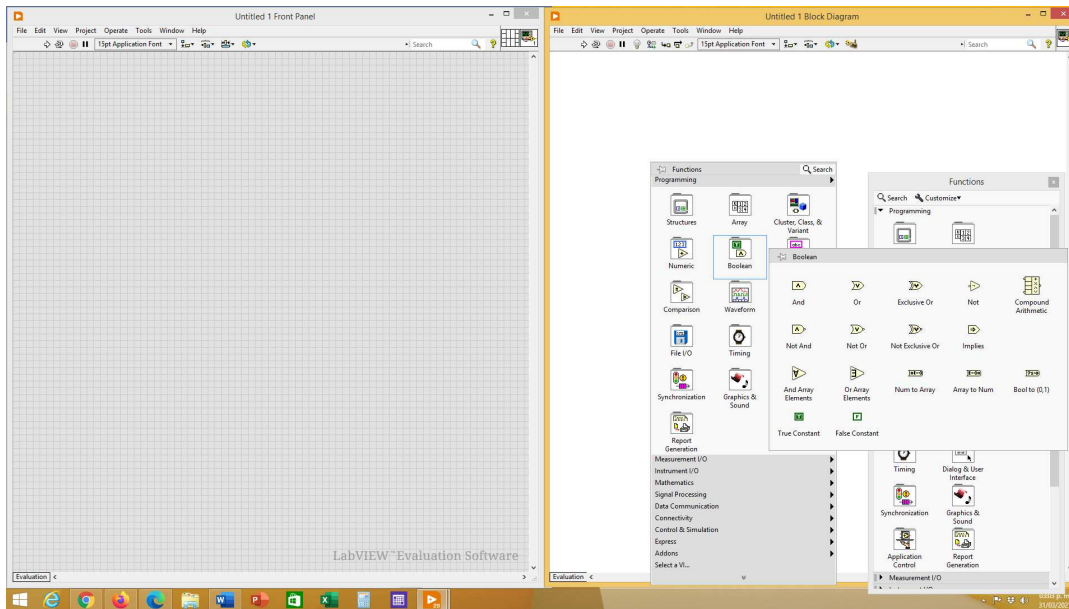
Otros bloques como Cluster, Class & Variant:



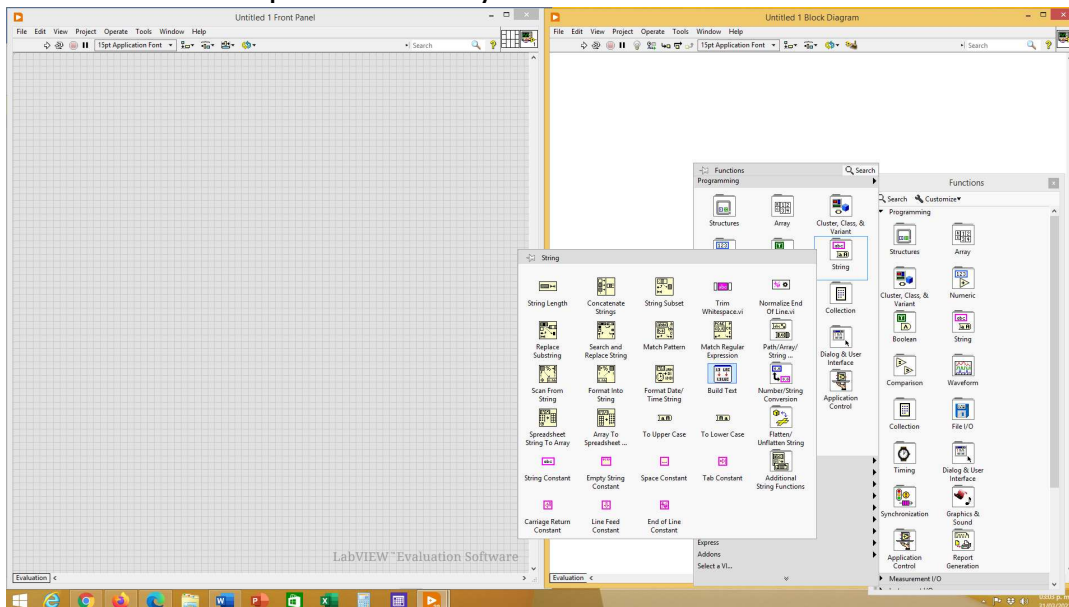
En la parte de bloques numericos, encontramos operadores como suma, resta, incrementos, raiz cuadrada, etc, y algunos controles para ingresar valores como constantes



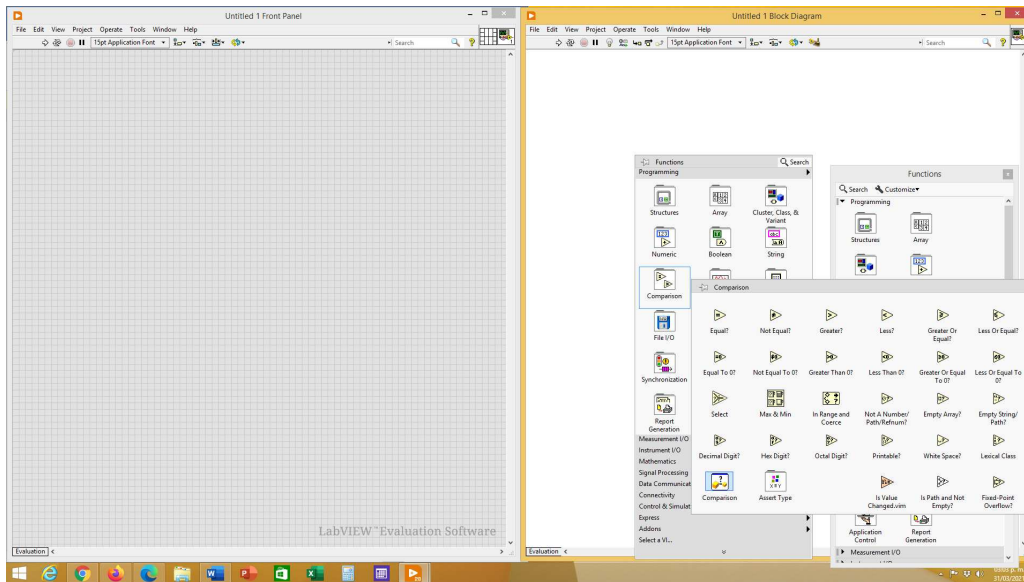
En el bloque de booleanos, tenemos las uniones logicas AND, OR, NOT, entre otras



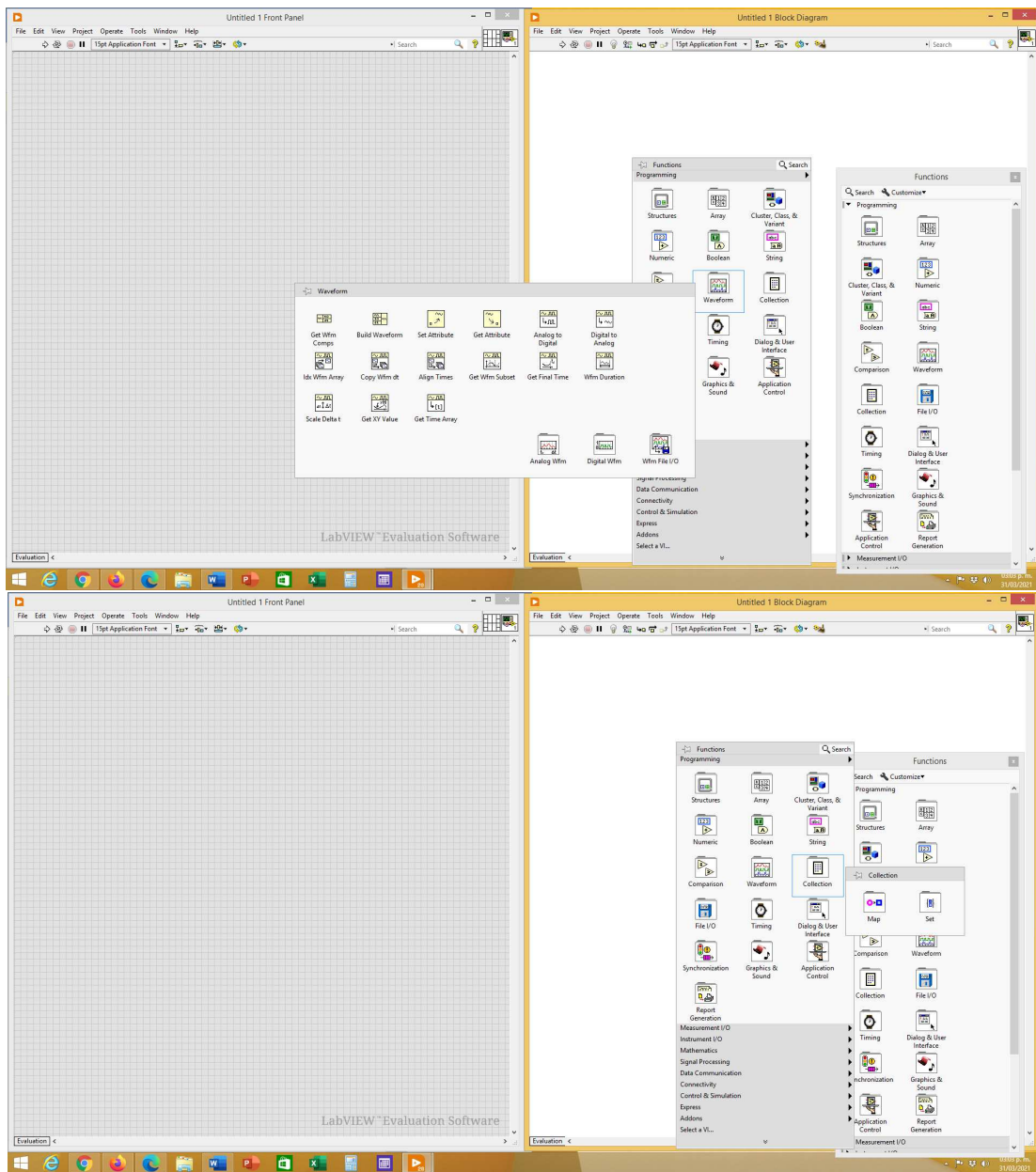
En el bloque de Cadenas o String, encontraremos bloques para constantes, convertir entre tipos de datos y mas.

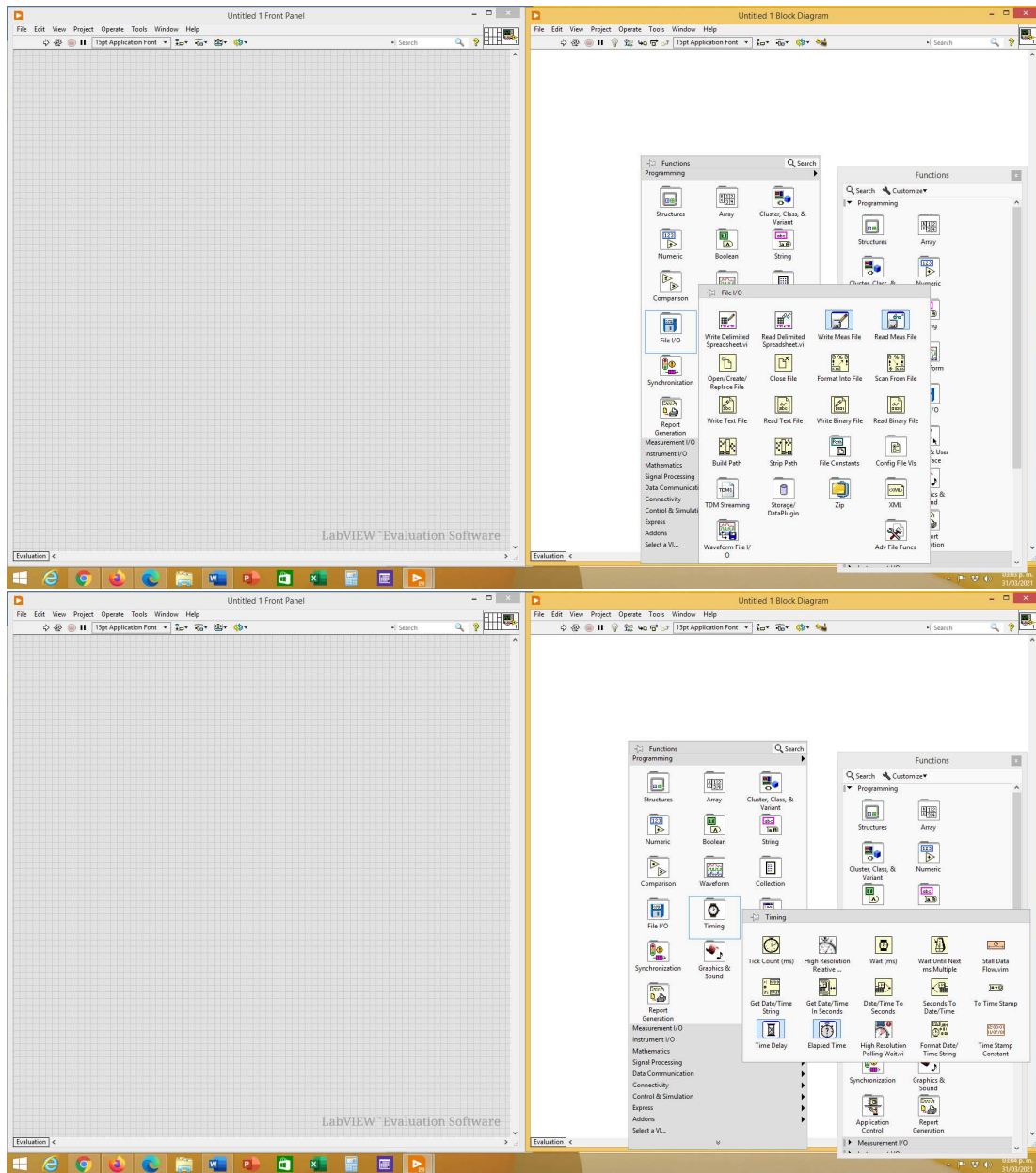


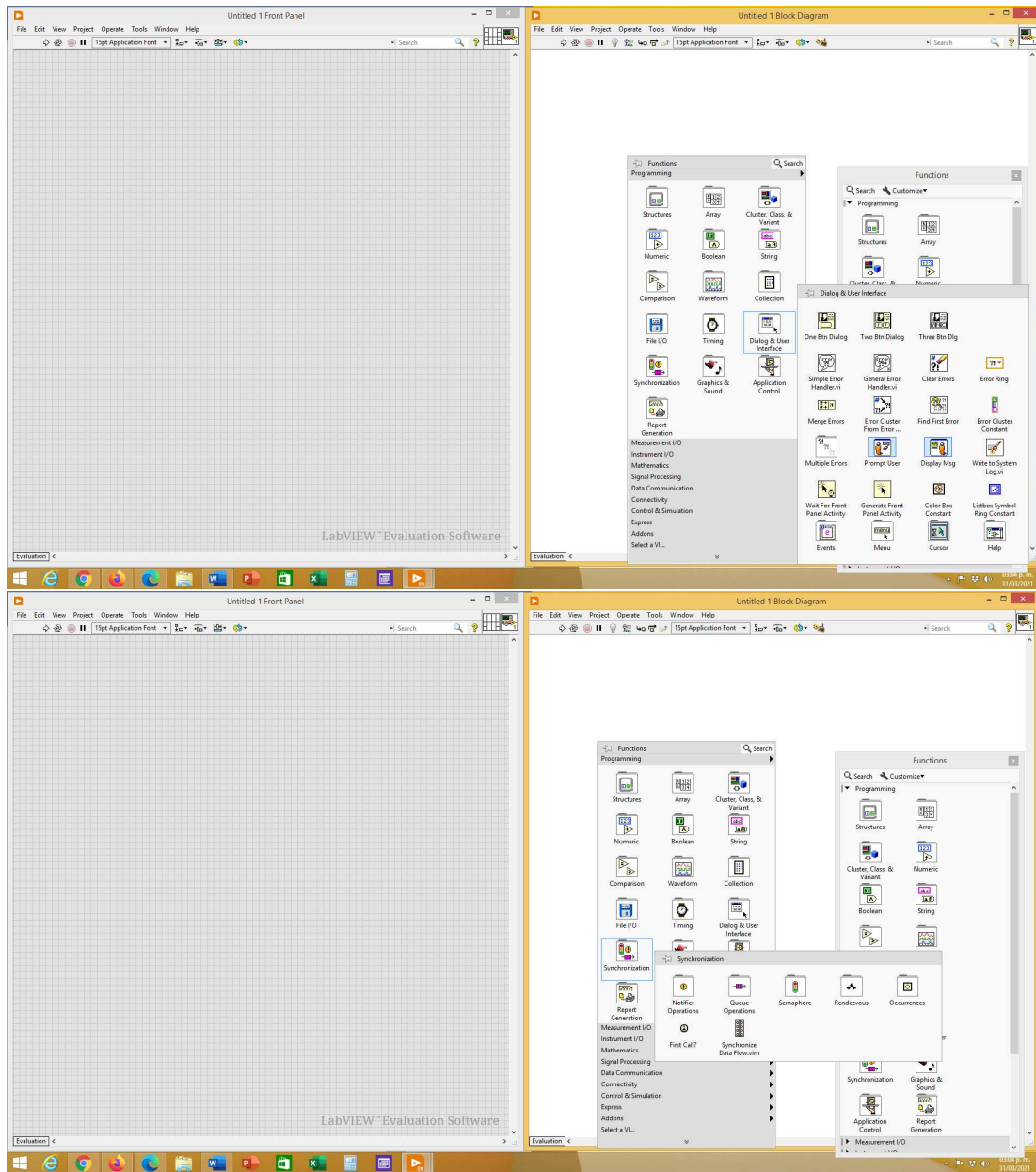
Otro menu de bloques muy empleado en las aplicaciones, es el de Comparadores, donde podemos comparar valores como se muestra en la figura.

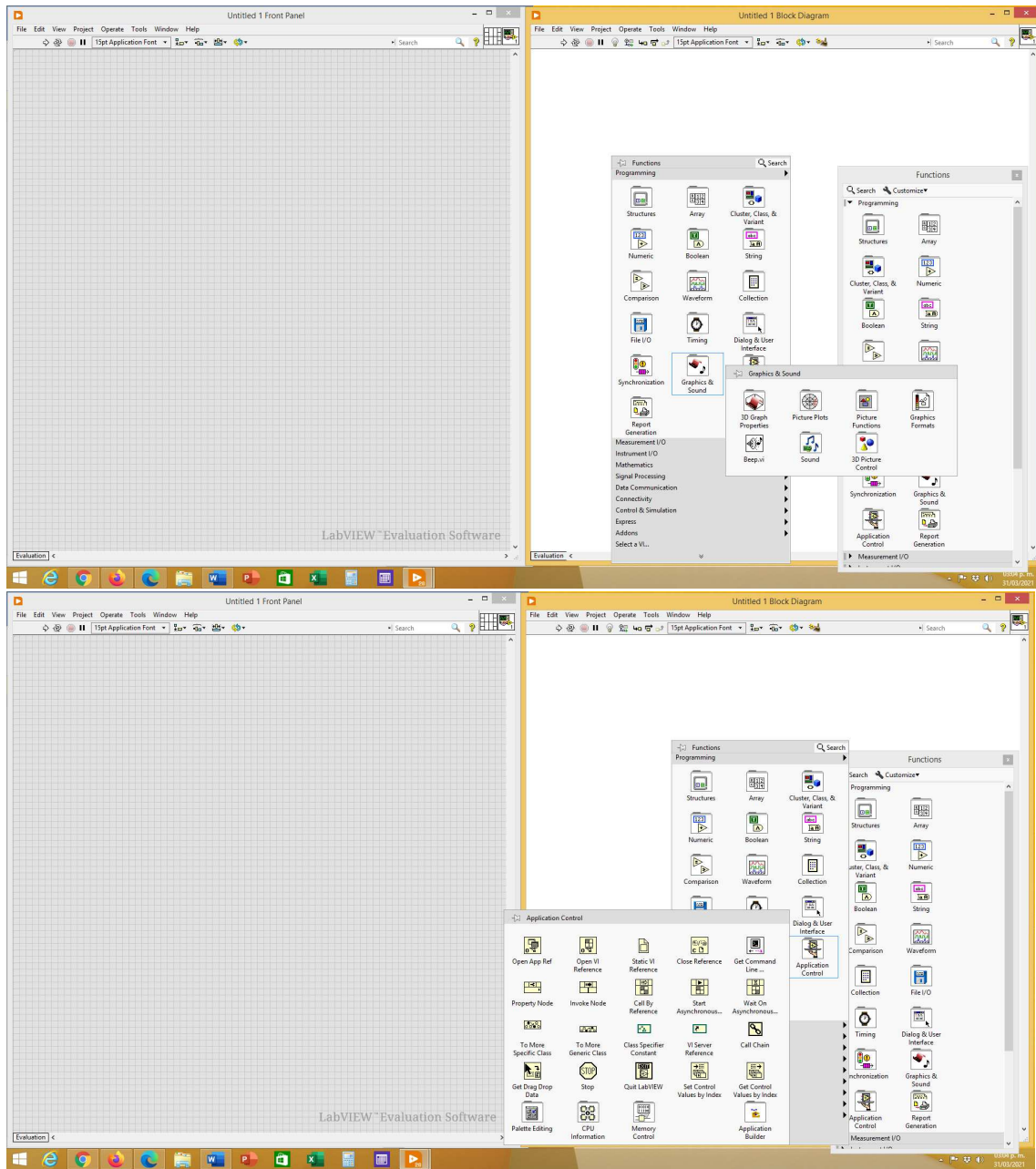


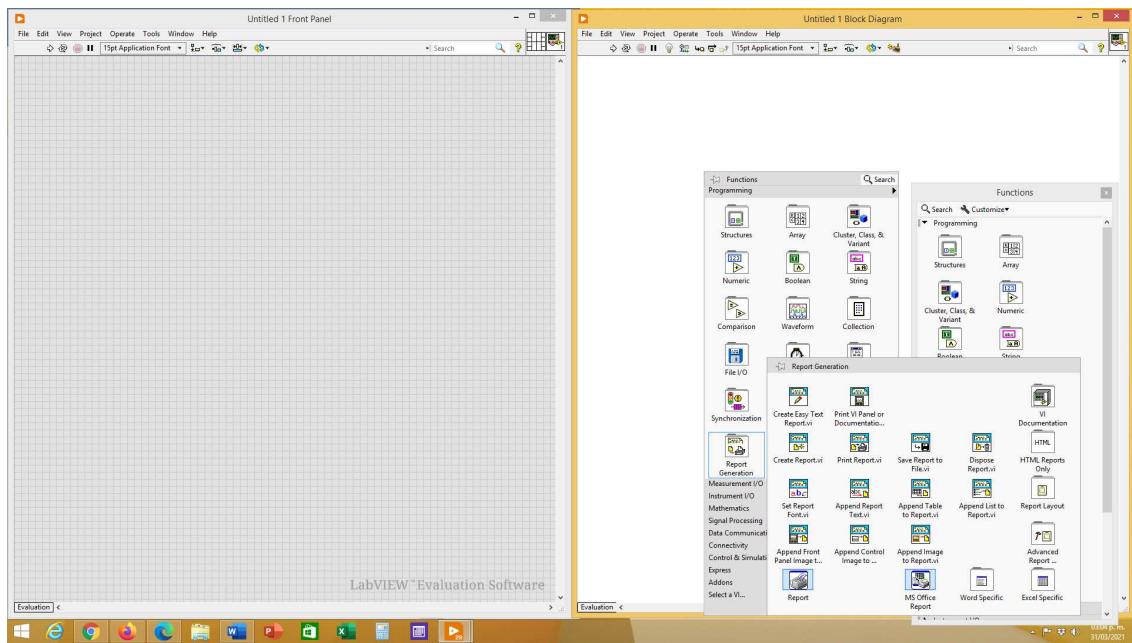
Y se muestran en las demás imágenes las opciones que encontrás en los demás bloques como referencia.





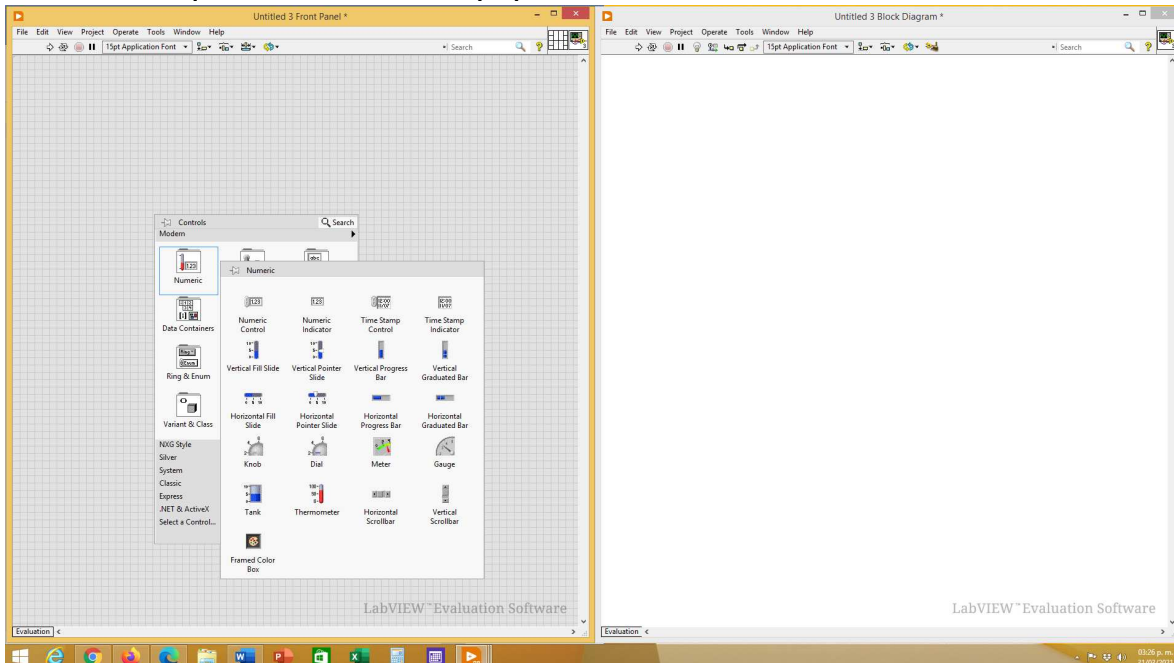






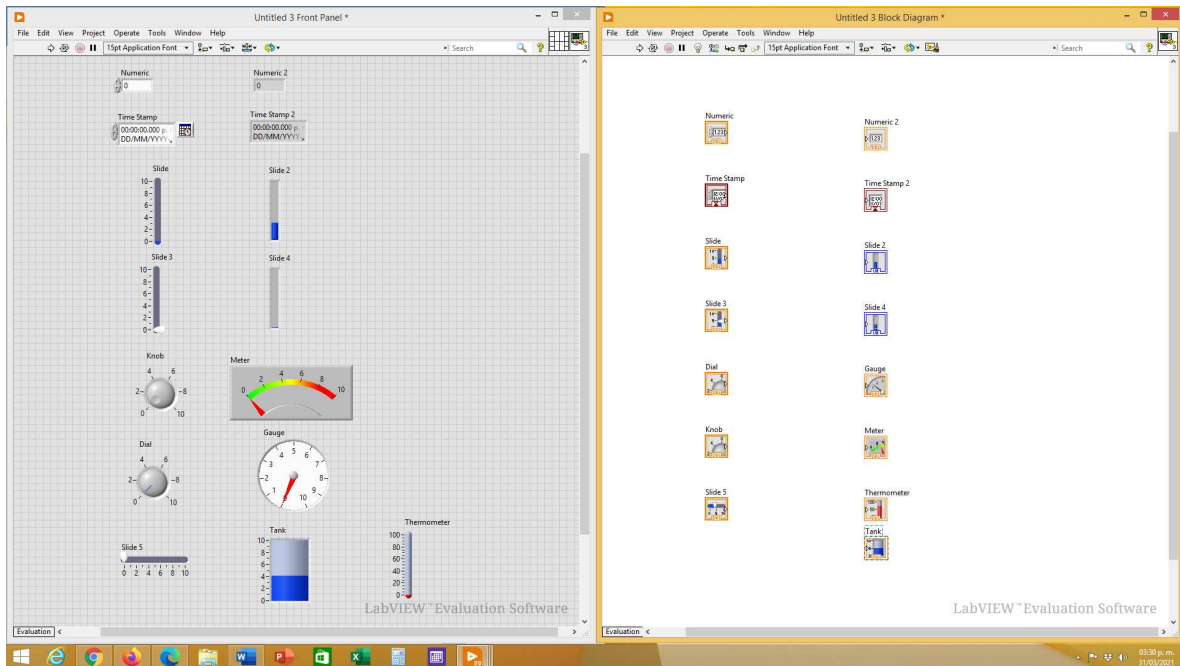
PRACTICA 3 ENTRADAS SALIDAS EN LABVIEW

Abrimos un nuevo proyecto o Blank VI, seleccionamos los controles numericos que sean entradas y que sean salidas, dando clic en el control.



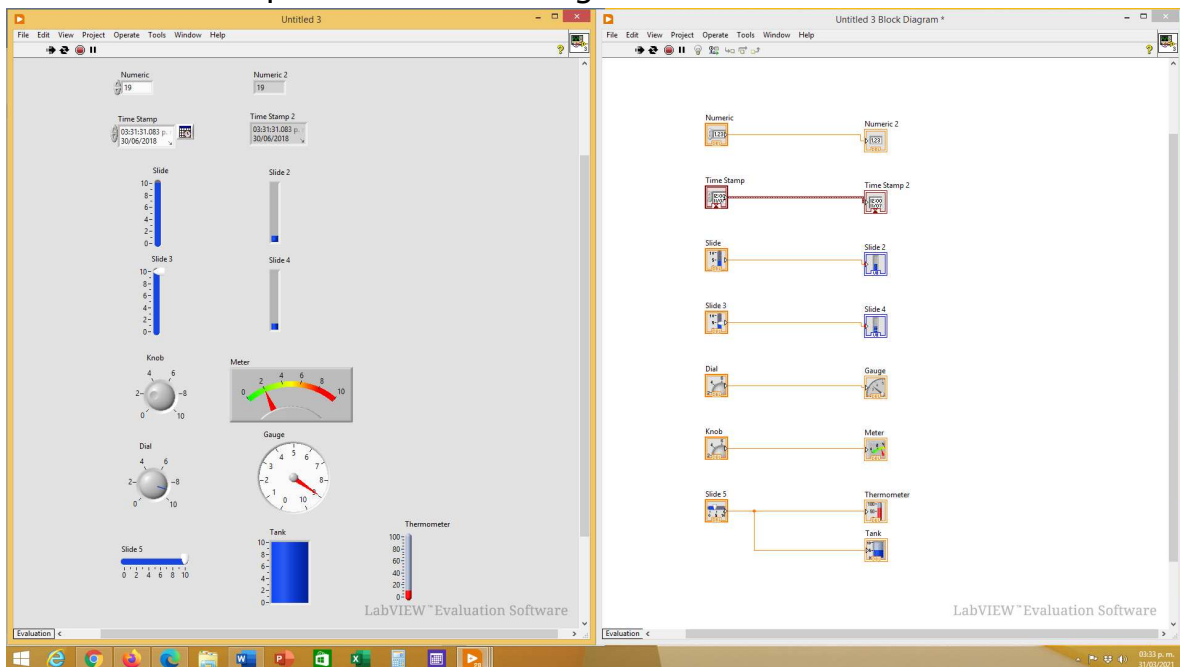
De lado de los Bloques, apareceran los iconos de bloque correspondientes. Para saber si un control es para mover, girar o presionar (ENTRADA), este bloque tendra una conexión del lado derecho.

Si un control es para Monitorear, como por ejemplo, un indicador de nivel de liquido, entonces, este bloque tendra un conector de lado izquierdo.



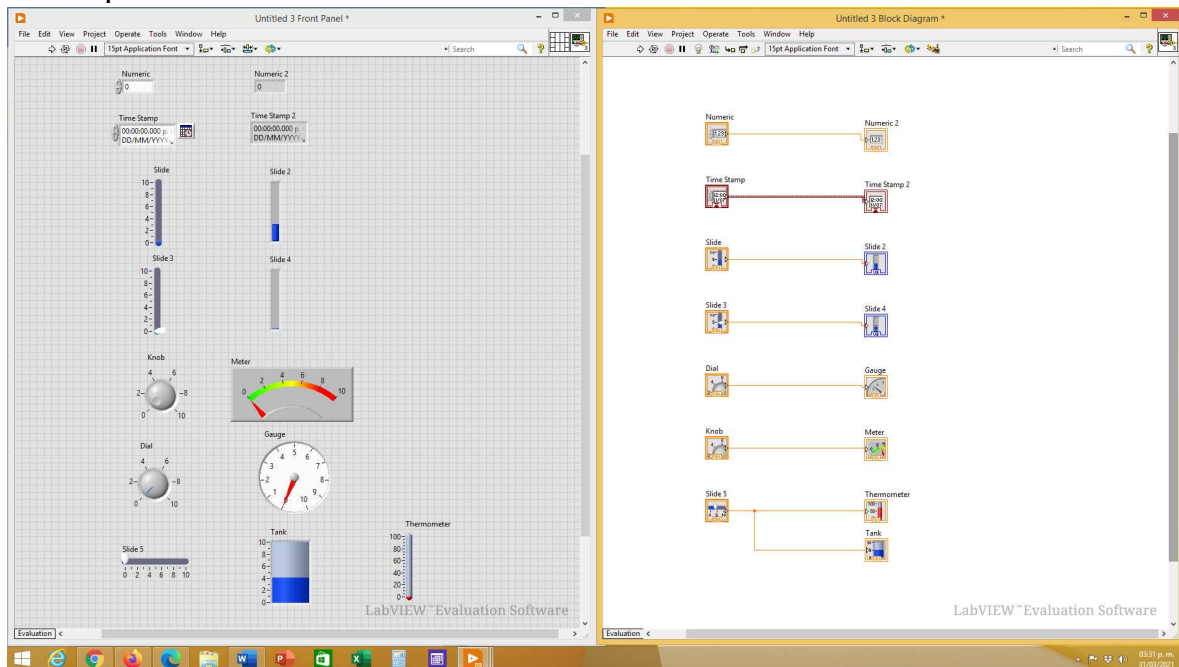
En esta practica, se conectaran Entradas con salidas directamente, y al ejecutar podemos ver el resultado de la interacción entre el control y el instrumento de monitoreo.

Las conexiones quedarian como la figura

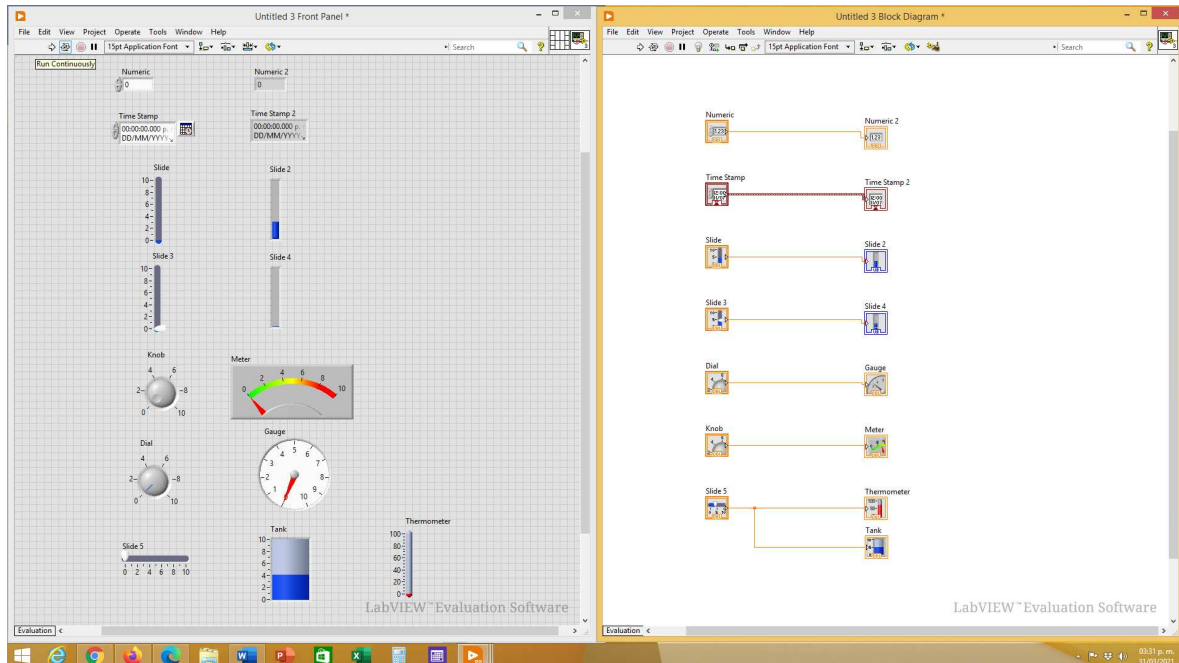


Si ejecutamos dando clic en las flechitas blancas en forma de circulo, y movemos la posicion de un control o cambiamos el valor, entonces del

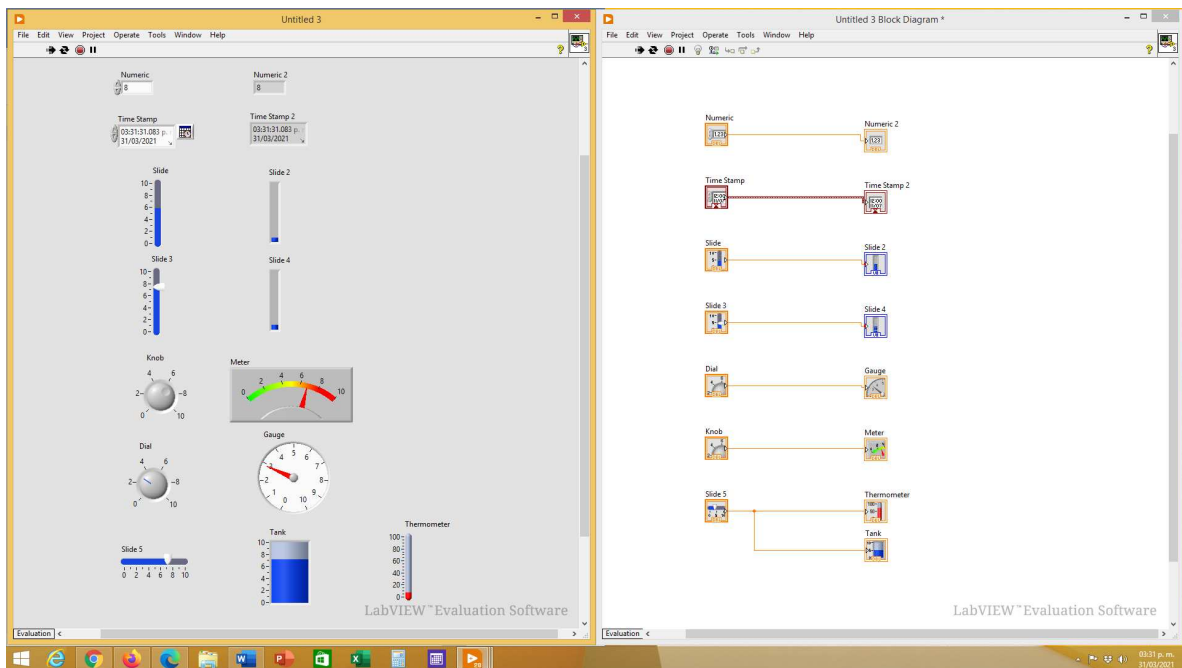
lado de los instrumentos de monitoreo, deberemos ver las respuestas correspondientes.



Observamos que tenemos controles de numeros, fechas, horas, posicion, etc.

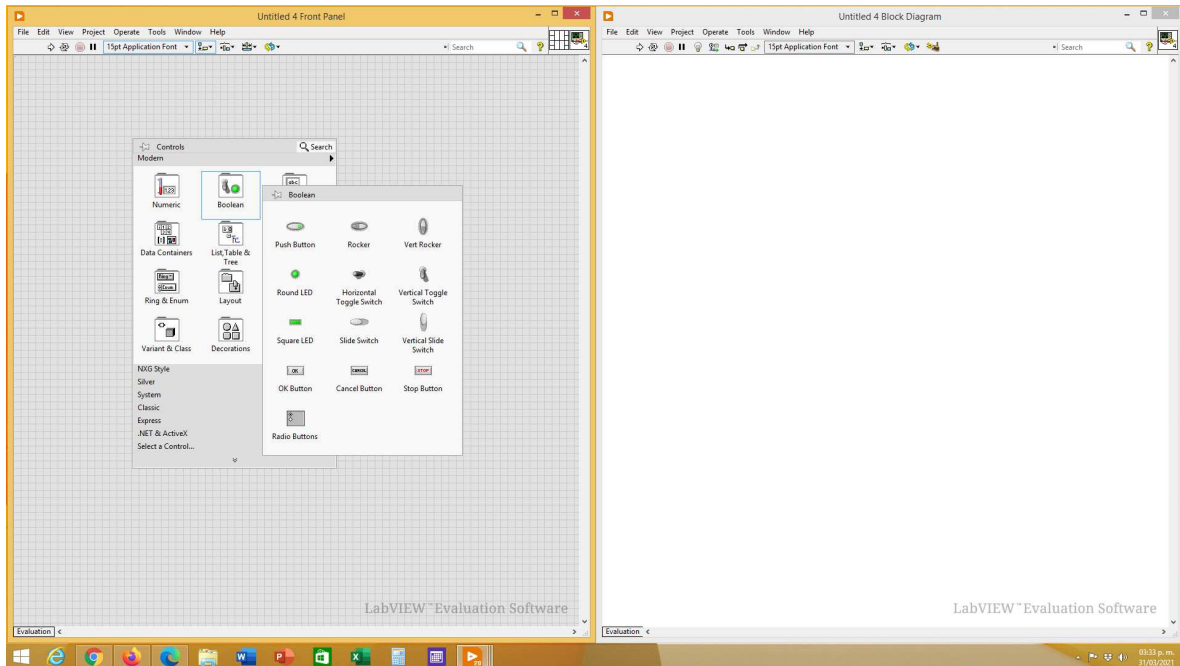


Y tambien observamos que tenemos indicadores que monitorean las entradas en forma de niveles, barras, numeros, agujas, etc

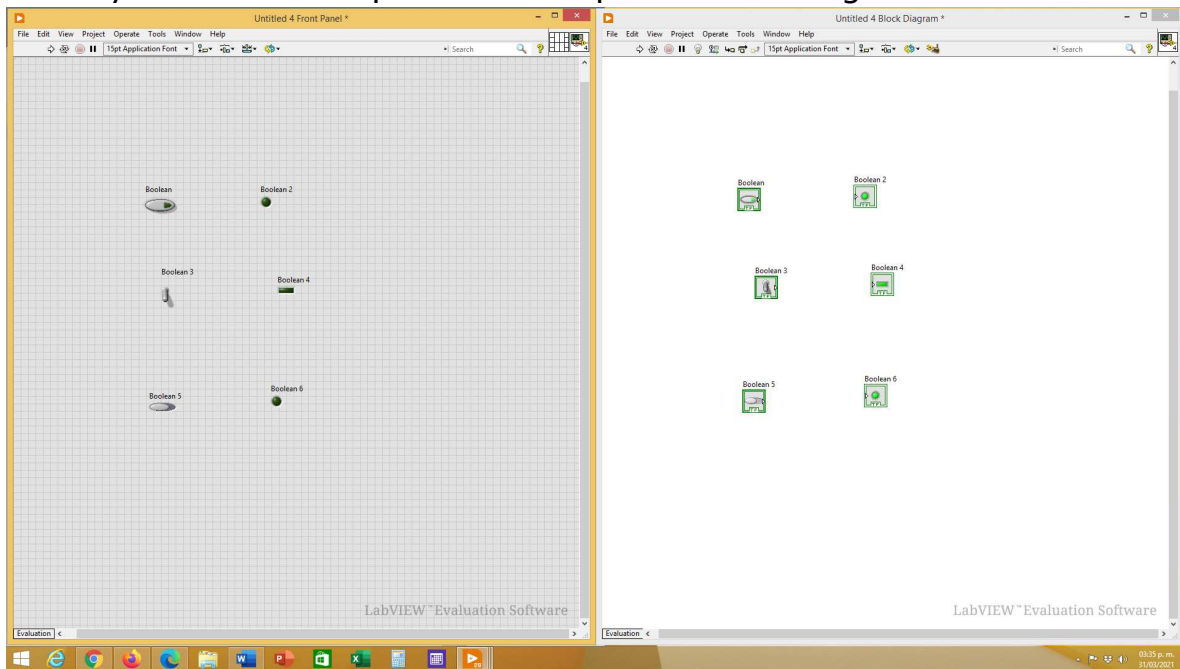


PRACTICA 4: ENTRADAS Y SALIDAS DIGITALES O BOOLEANAS.

Al igual que el ejercicio anterior, en este caso iremos la menu de controles Booleanos.

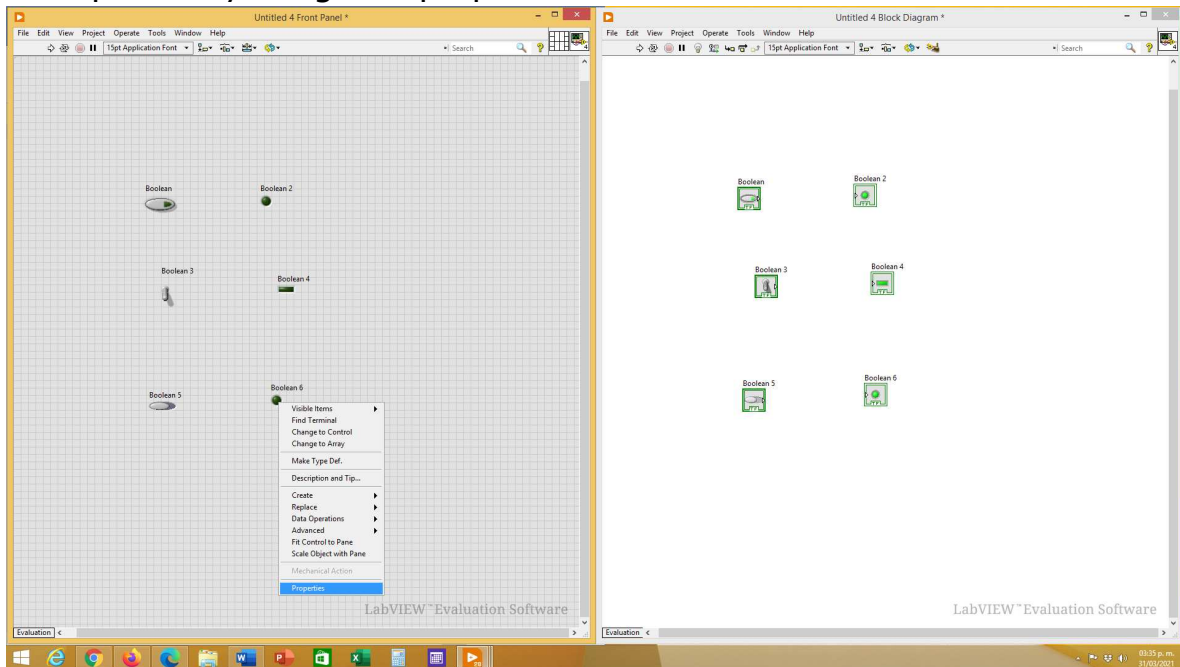


Ingresaremos botones o interruptores como entradas y leds como salidas. Los hay de diferentes tipos como se aprecia en la imagen

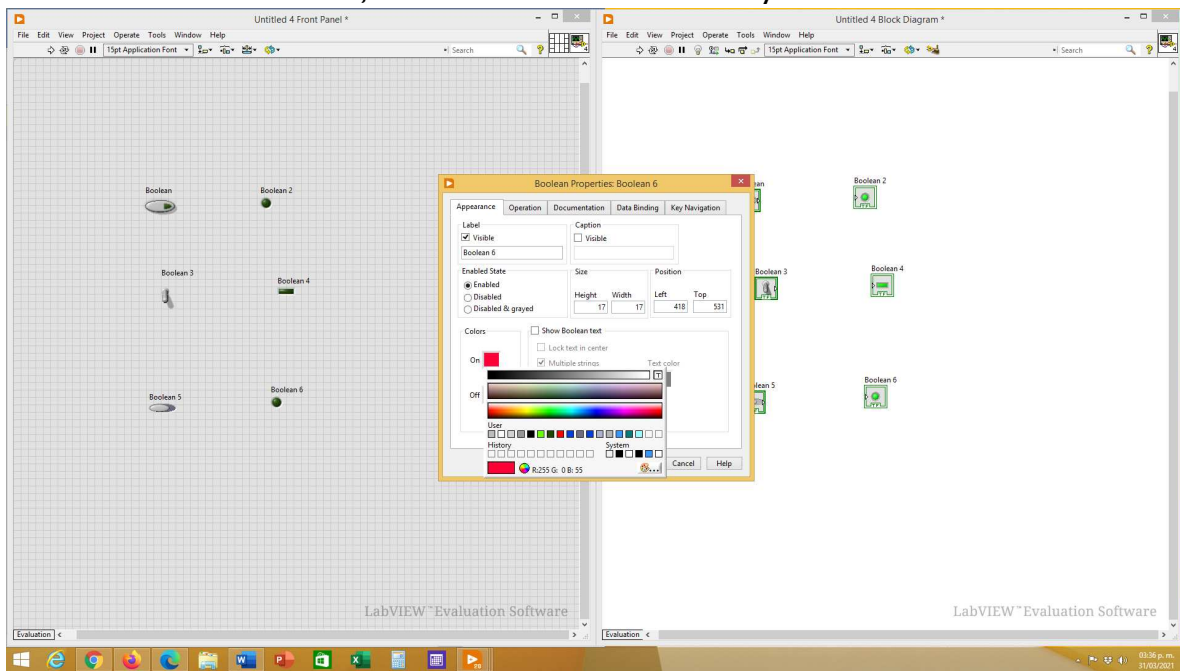


En la ventana de los bloques, apareceran los iconos correspondientes.

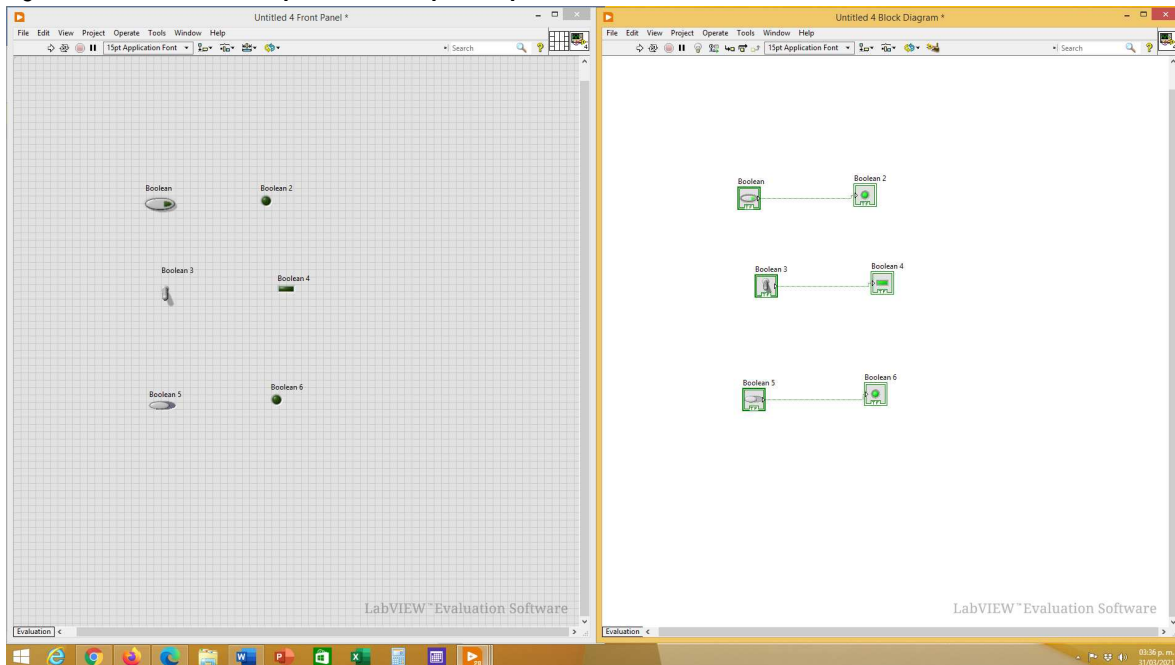
Si son entradas, tendran un conector del lado derecho.
 Si son salidas, el conector estara del lado izquierdo.
 Para cambiar el color de algun led, debemos de dar clic con boton derecho al dispositivo y elegimos propiedades.



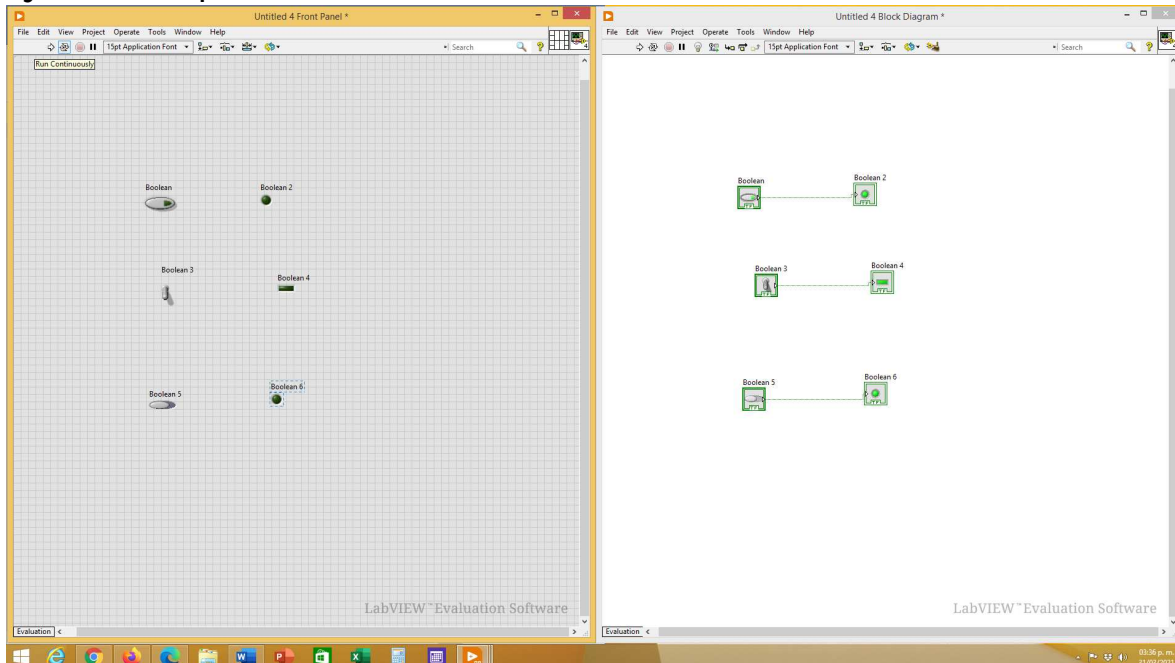
Aparecera una ventana con la pestaña Appearance o apariencia y ahí podemos cambiar el color del led cuando este apagado y cuando este encendido. Por default, el color es verde oscuro y verde claro.

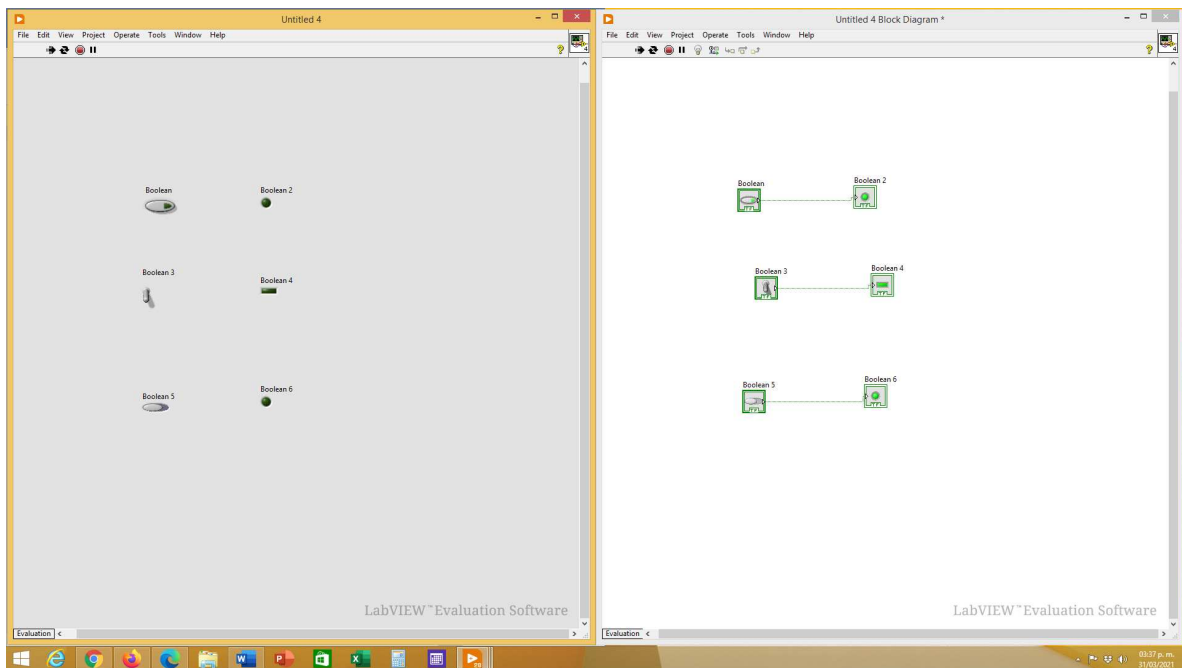


Conectamos los bloques de entrada y salida y posteriormente ejecutaremos la aplicación para probarlos.

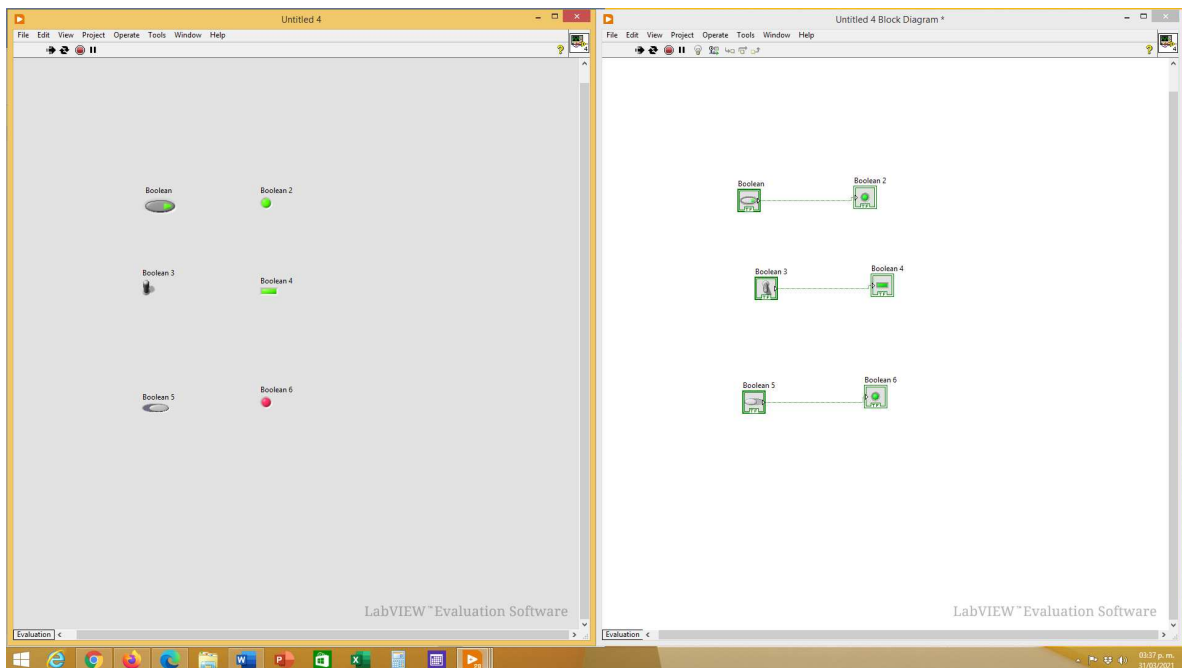


Buscamos el icono que tiene dos flechas blancas en forma circular para ejecutar la aplicación.





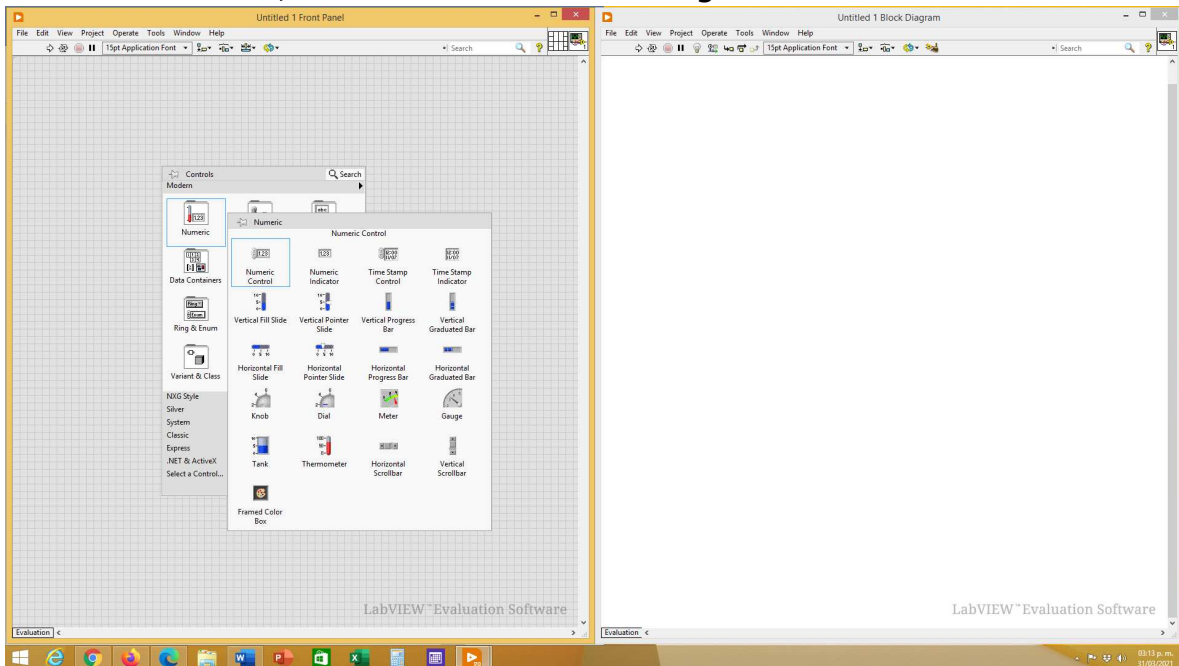
Ahora interactuamos con los botones y veremos como encienden y apagan los leds



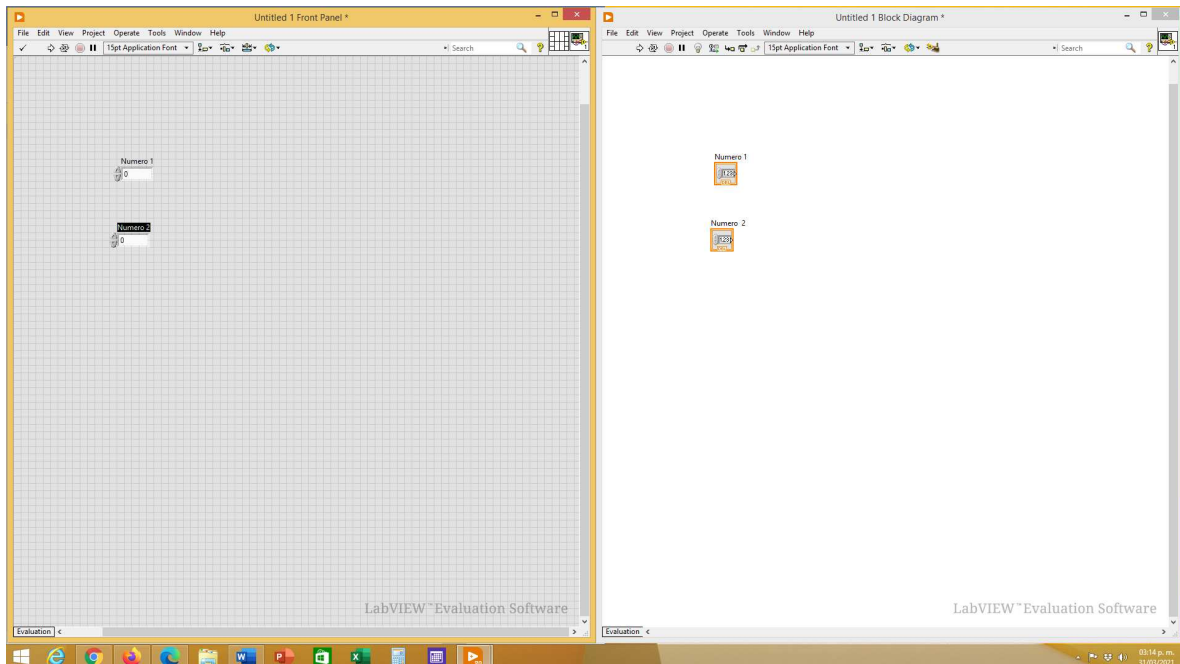
PRACTICA 5: Suma de dos numeros usando controles numericos con LabView

Abrimos un proyecto nuevo dando clic en Blank VI

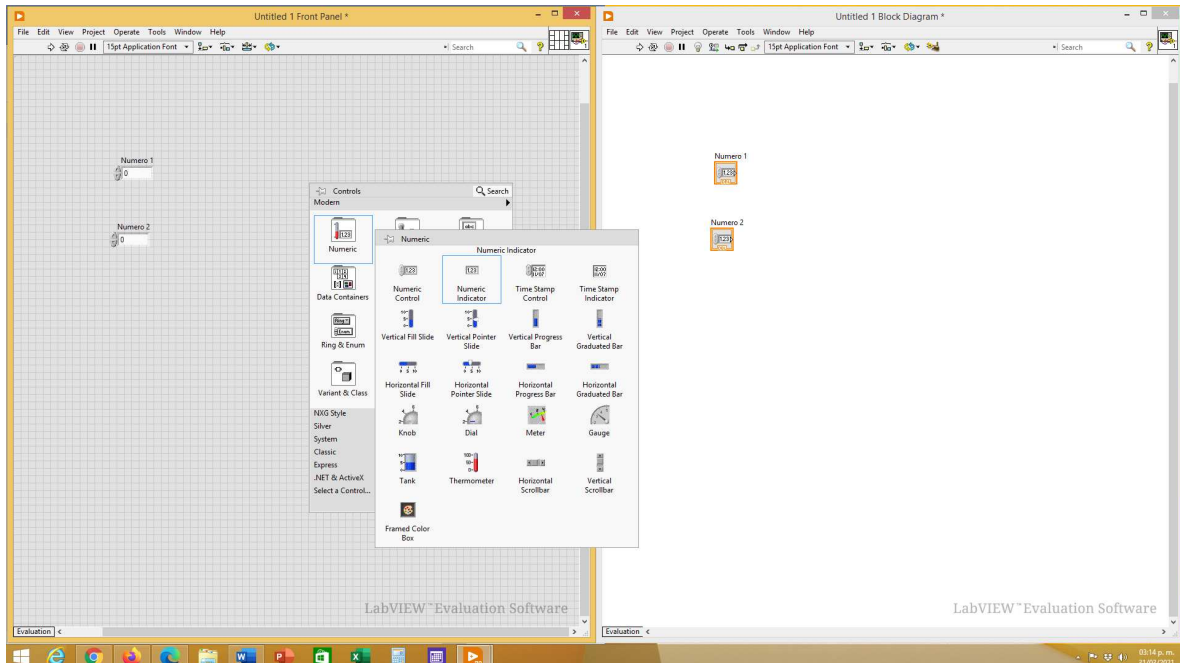
Despues clic con boton derecho para agregar en el Menu Numeric, un control Numerico, esto se muestra en la imagen



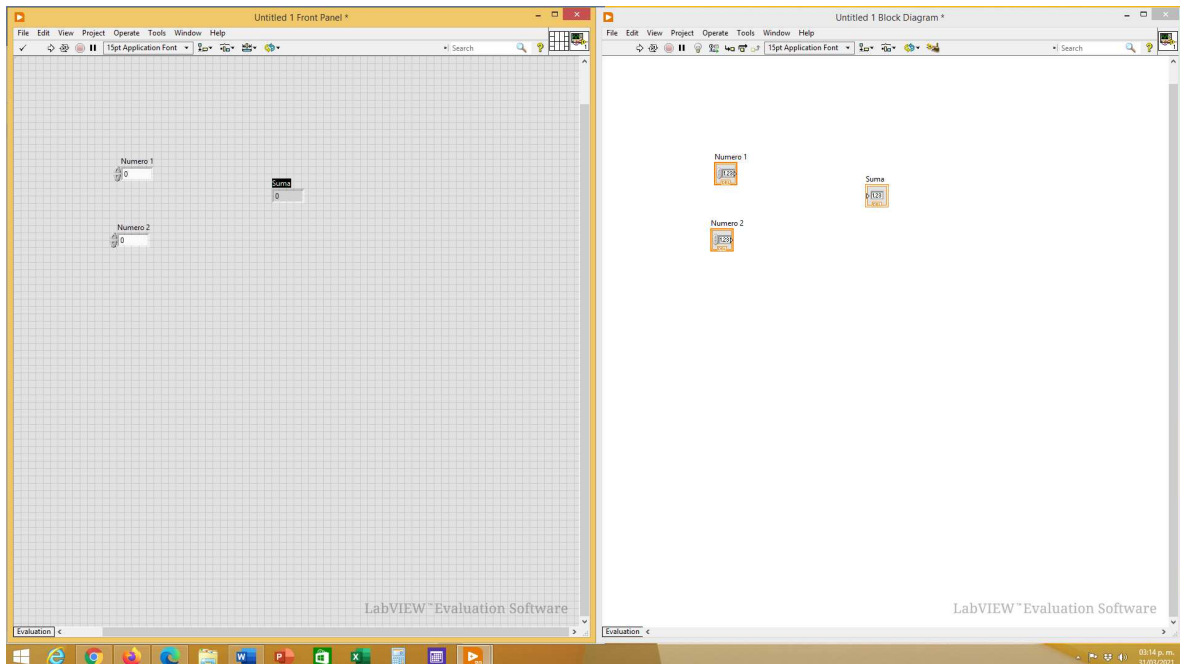
Del lado de los bloques aparecera el bloque de este elemento. Agregamos un control más para tener 2 entradas de numeros. Estas entradas serán los controles donde colocaremos los numeros a sumar:



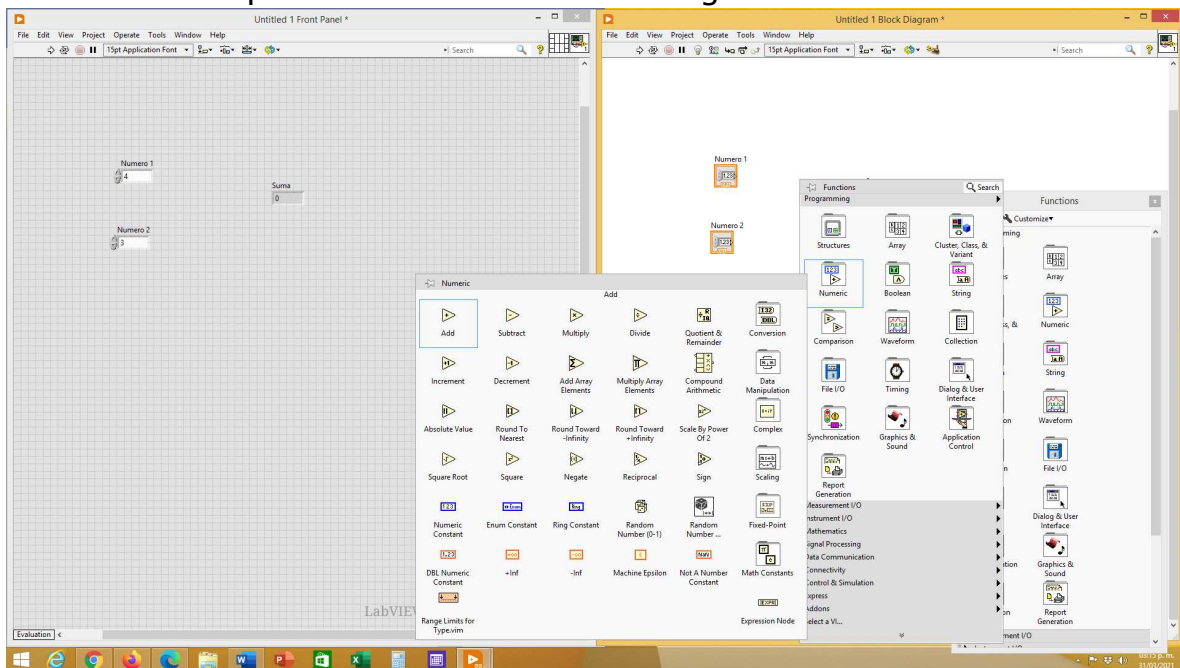
Posteriormente, metemos un control para salida de resultados. Una vez que se sumen los numeros, aquí se mostrara el resultado. Este control aparece igual en el menu Numeric, y consiste en el control llamado Numeric Indicator



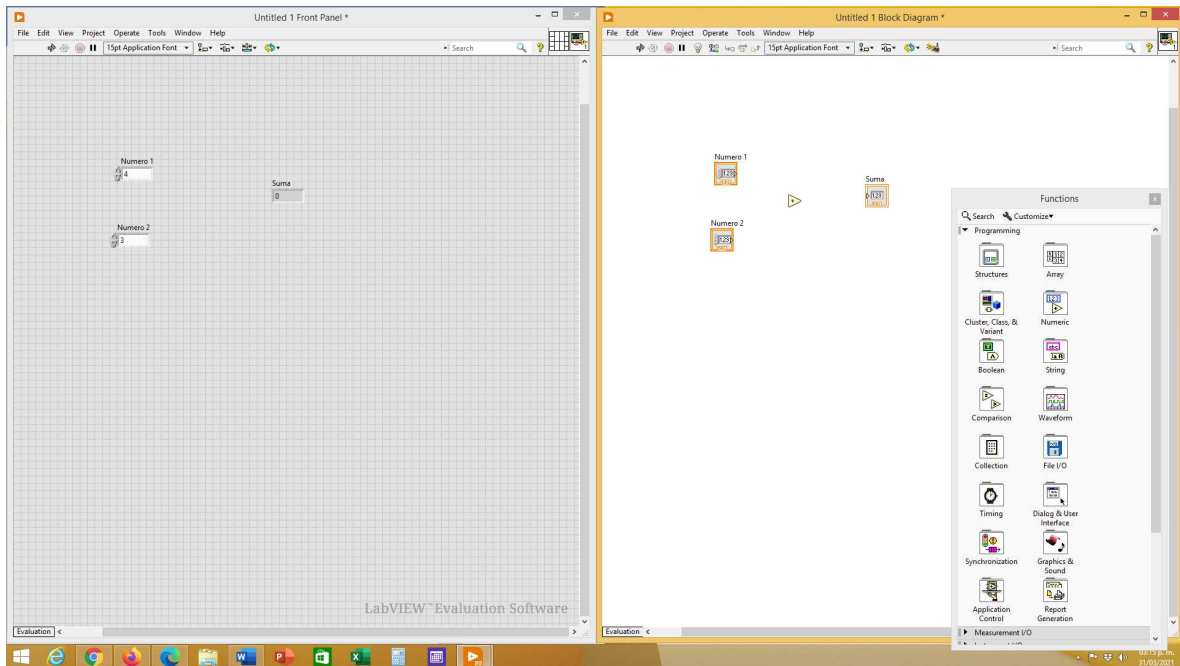
Las etiquetas de cada control se pueden cambiar (textos) dando doble clic sobre el mismo texto



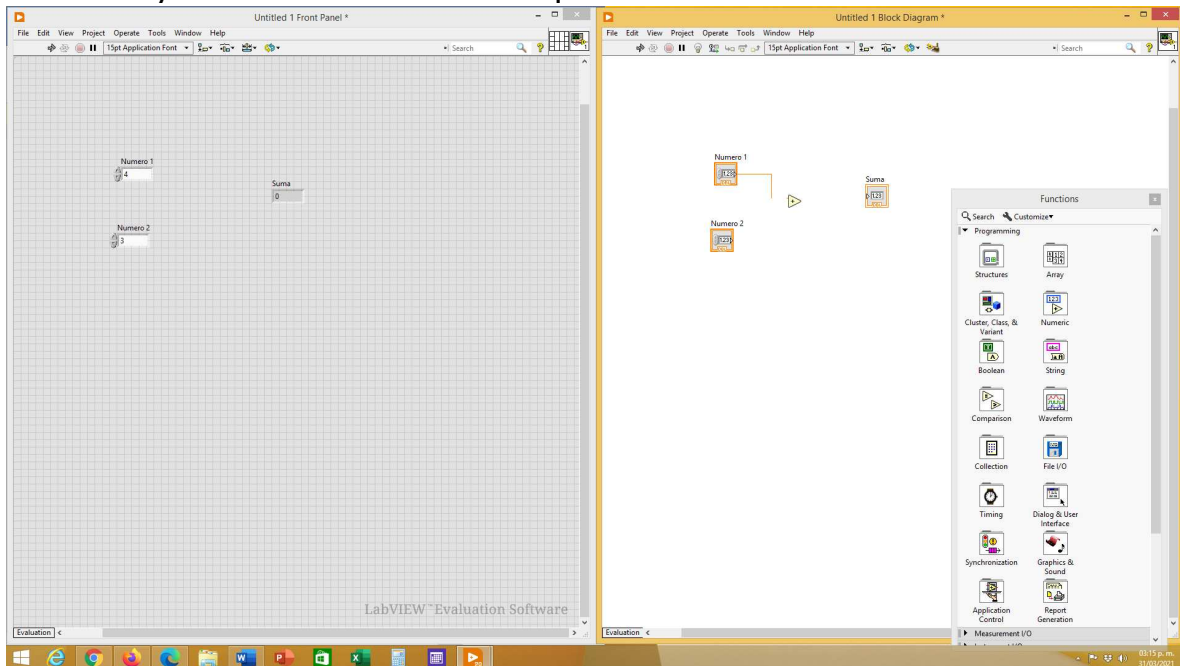
Procedemos a agregar un control para la suma. Esto lo metemos en la ventana de Bloques tal como lo indica la imagen.



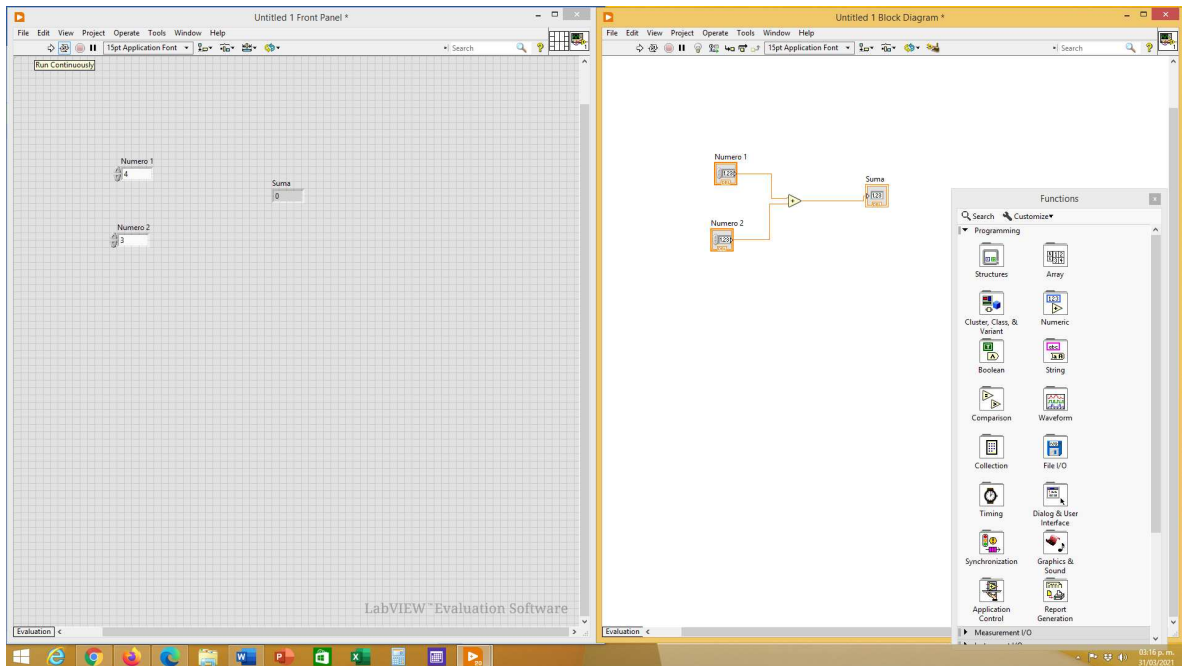
Depues quedaria asi:



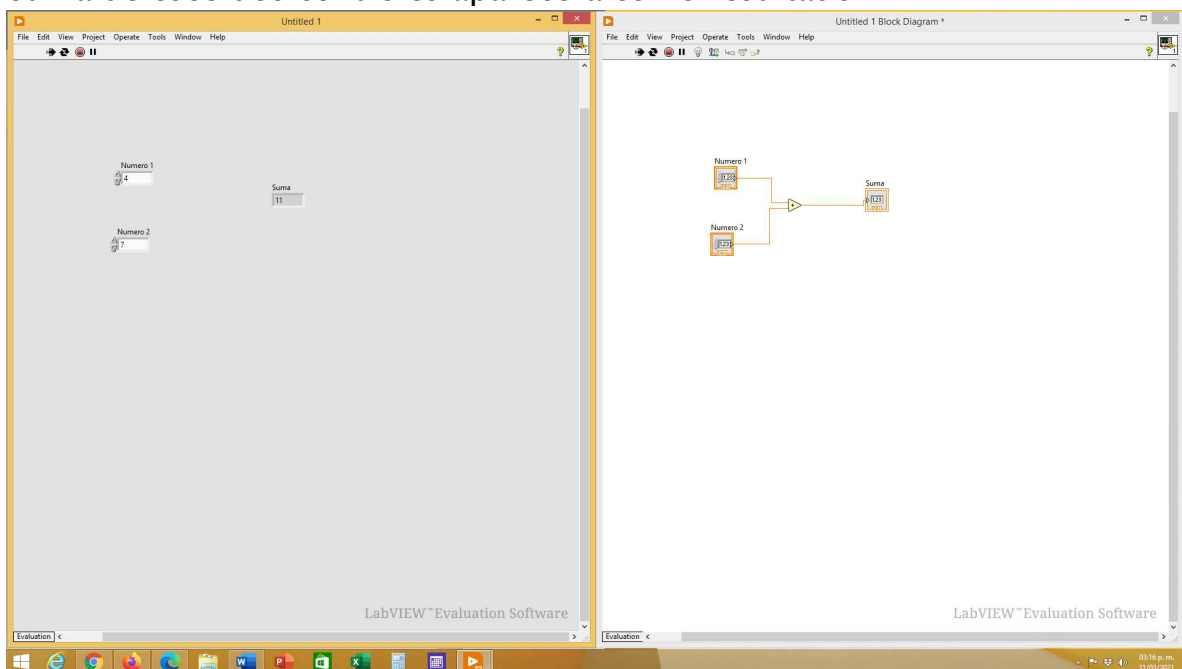
Lo siguiente seria conectar cada bloque de numero con las entradas del sumador y la salida con el control que mostrara el resultado.



La conexión seria la siguiente:



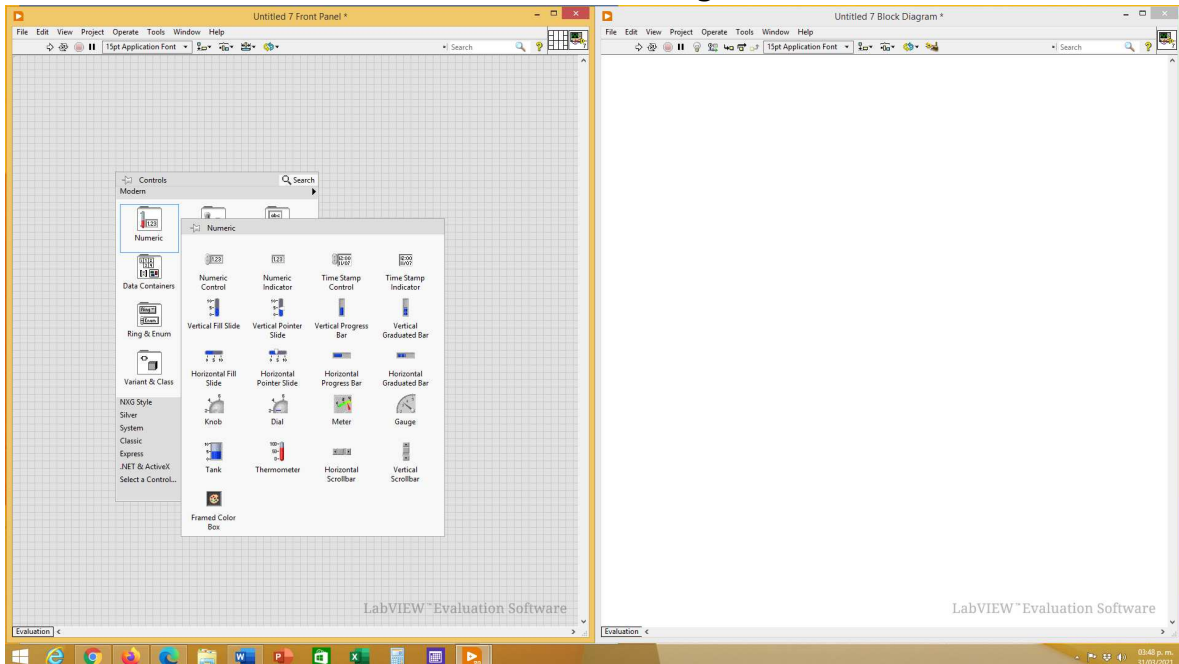
Para ejecutar, damos clic en las flechas blancas de forma circular y con las flechitas, cambiamos los numeros de las entradas. Observaremos que la suma de esos dos controles aparecera como resultado



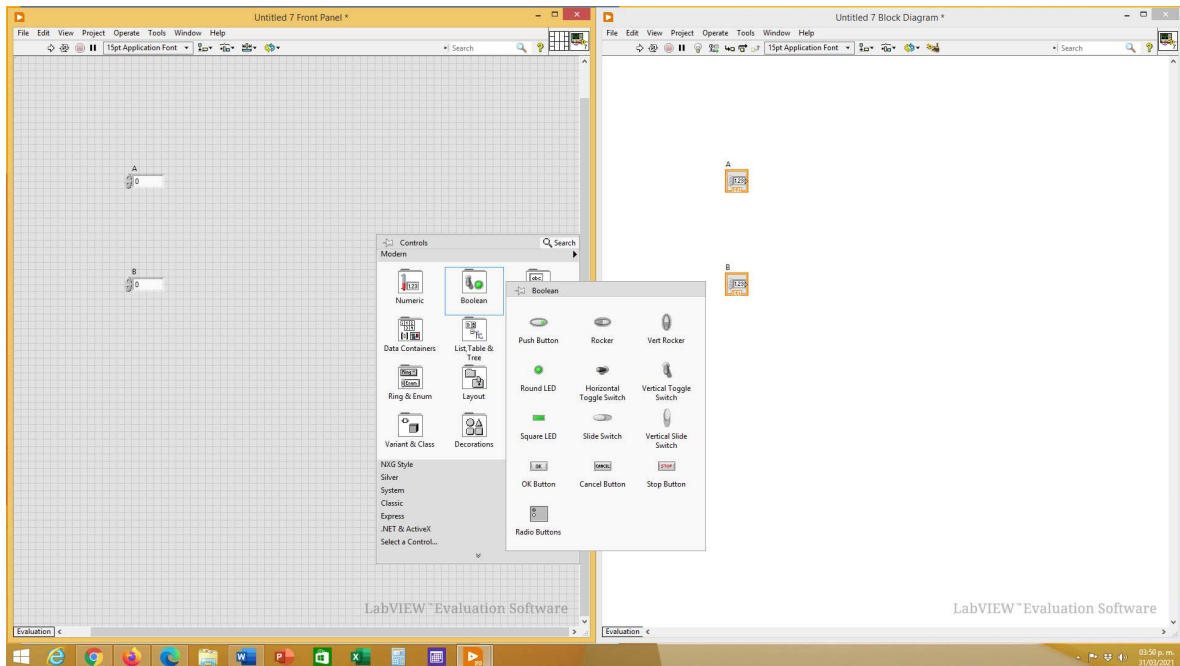
PRACTICA 6: Comparación entre 2 numeros y se muestra si son iguales o diferentes

Abrimos un proyecto nuevo dando clic en Blank VI

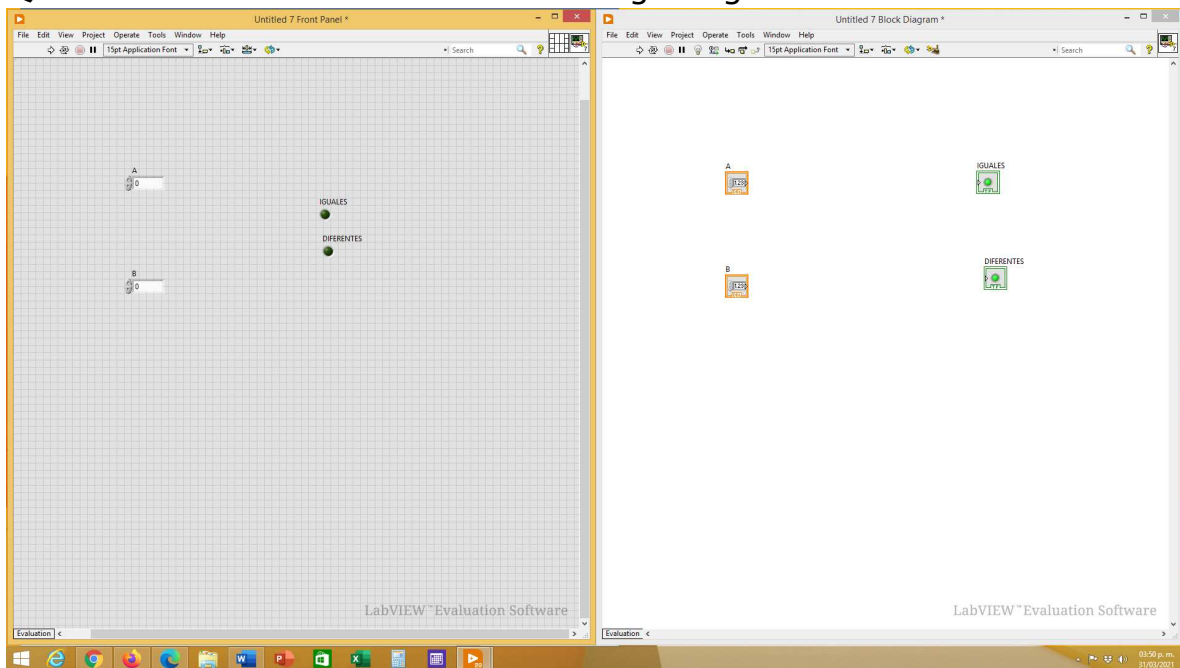
Despues clic con boton derecho para agregar en el Menu Numeric, un control Numerico, esto se muestra en la imagen



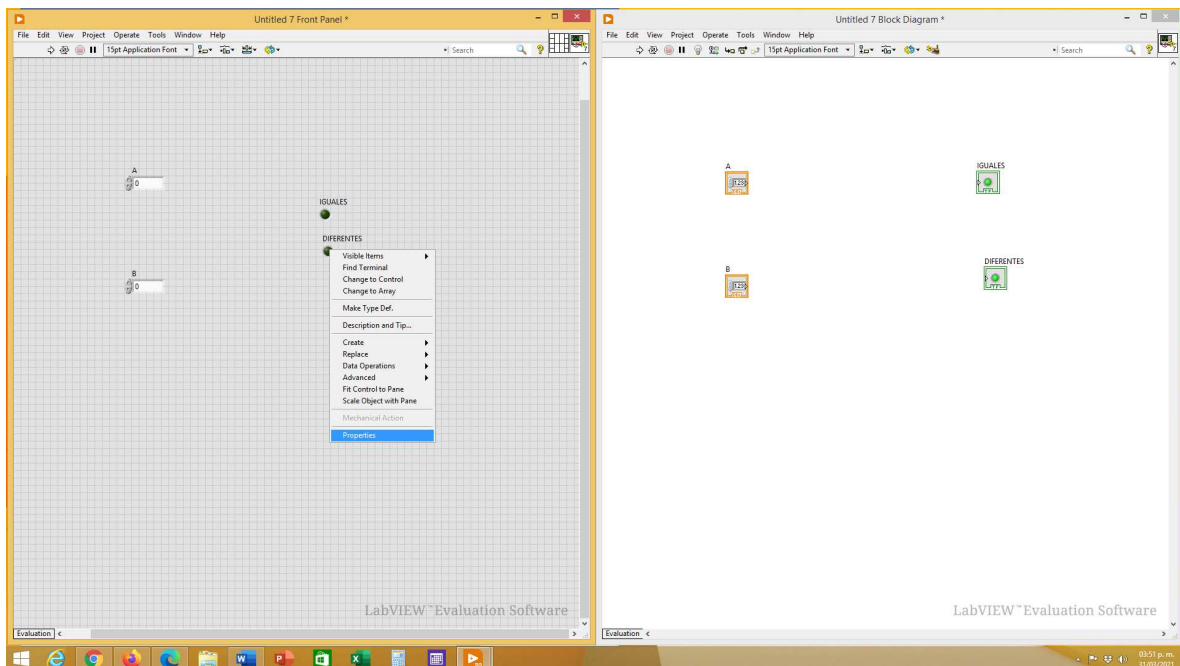
En total agregamos dos de estos controles y observaremos que se añadiran automaticamente sus bloques respectivos en la ventana derecha.
Despues agregaremos 2 leds. El primero encendera si los numeros son iguales y el segundo encendera cada vez que sean diferentes. Para agregarlos usamos los controles del menu Boolean



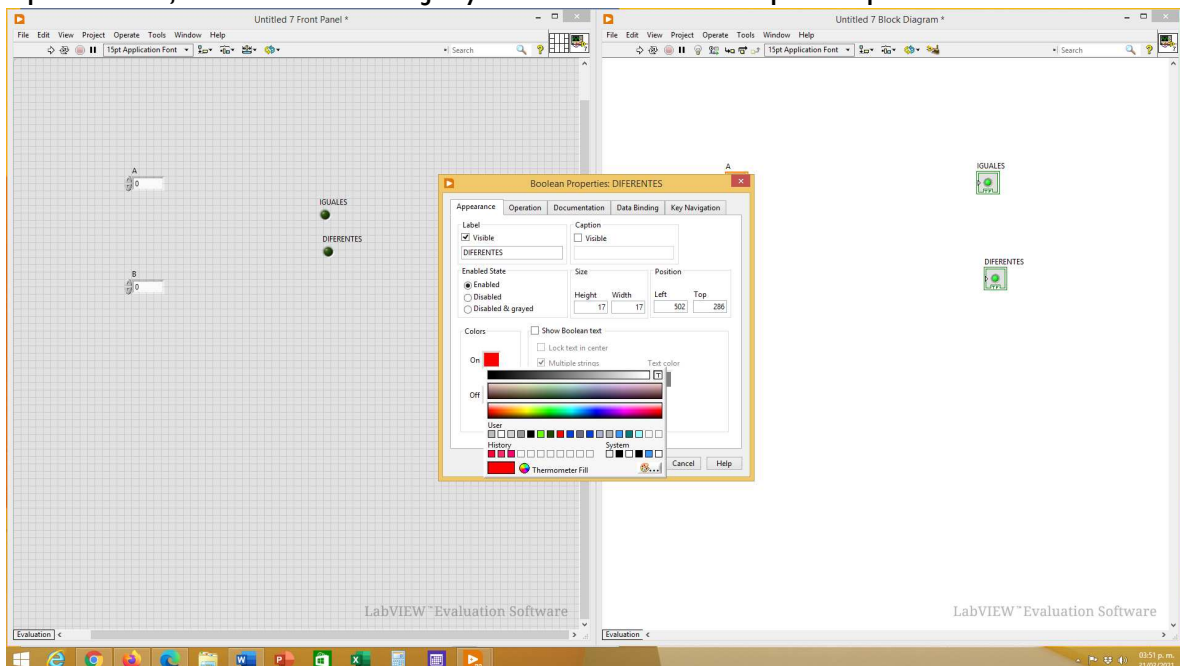
Quedaría así como se muestra en la imagen siguiente.



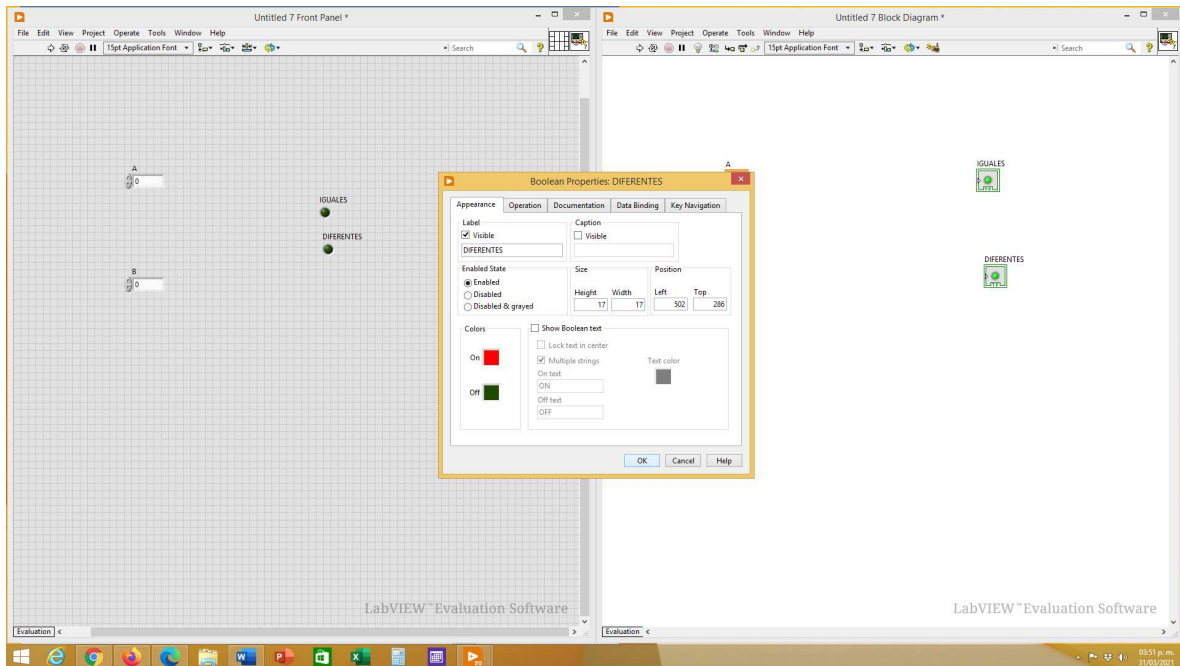
Para identificar los dos leds, cambiaremos el color del led que indique que son diferentes, colocando el color rojo cuando encienda. Para esto damos clic en propiedades



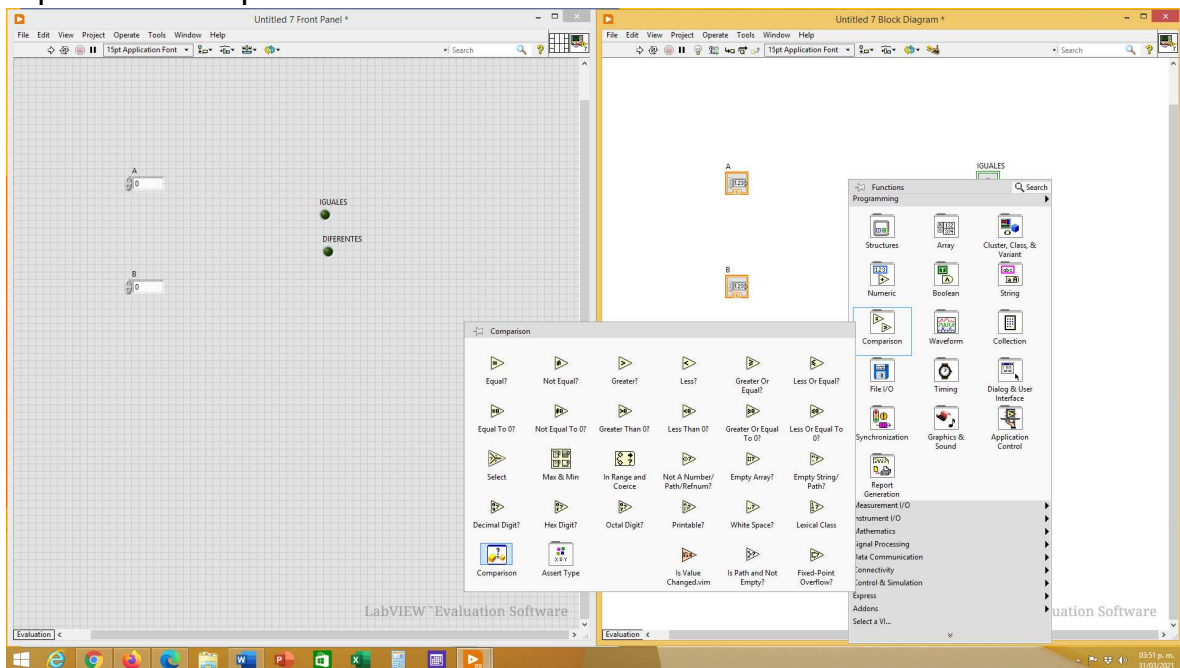
Despues con los colores disponibles en la pestaña Appearance o apariencia, buscamos el rojo y damos clic en OK para aplicar el cambio.



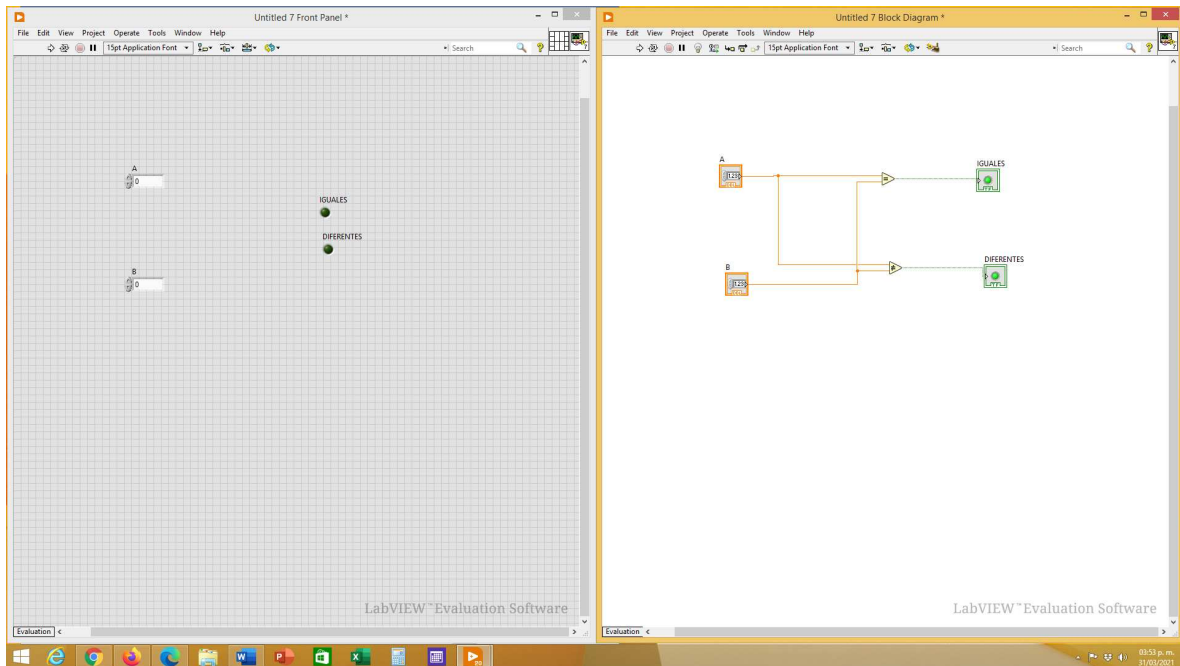
Se realizara solo en el led de diferentes, pero tambien se puede hacer con el led de iguales, procurando que ambos leds sean diferentes en color.



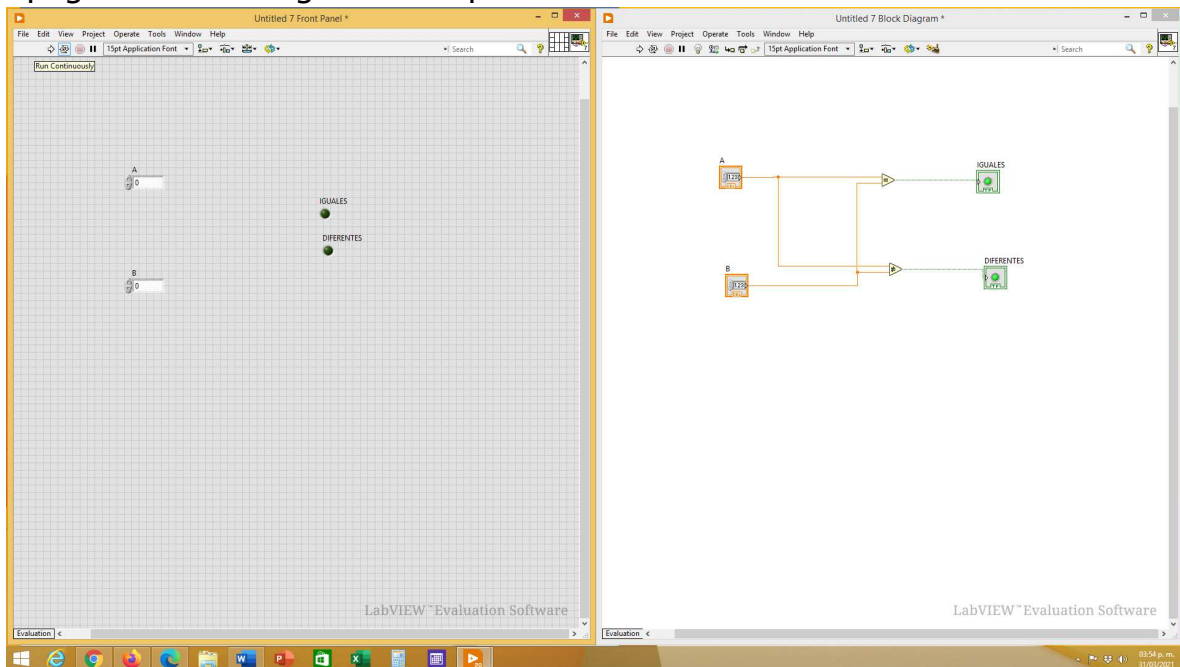
Despues vamos a ingresar el control de comparación eligiendo los bloques Equal? Y Not Equal?



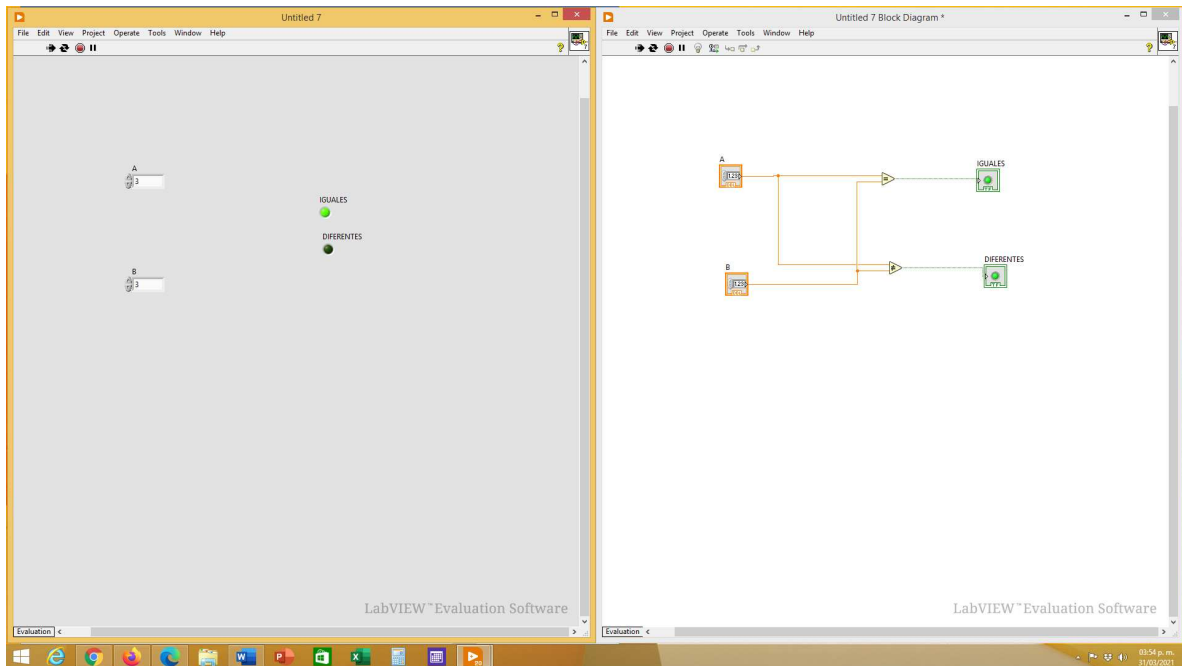
Los conectamos con las entradas numericas A y B como se muestra en la figura. Y la salida de Igual, la conectamos con el led verde (iguales) mientras que la salida de Diferentes, la conectamos en el led rojo.



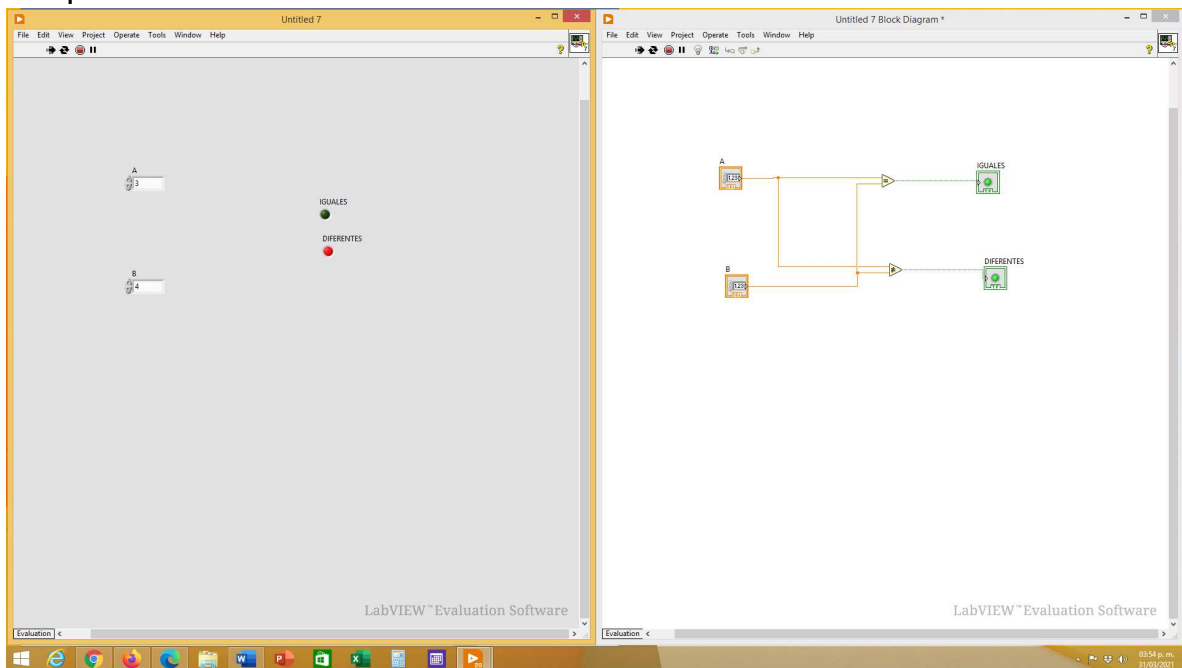
Eso seria todo, y a continuación, vamos a ejecutar dando clic en las flechas blancas de forma circular. Si apareciera algun error, ahí se mostraria. En caso de que todo este bien, podremos mover los controles de los numeros cambiando los valores y al mismo tiempo observando como encienden o apagan los leds según la comparación.



Aquí se muestra cuando los dos numeros son iguales



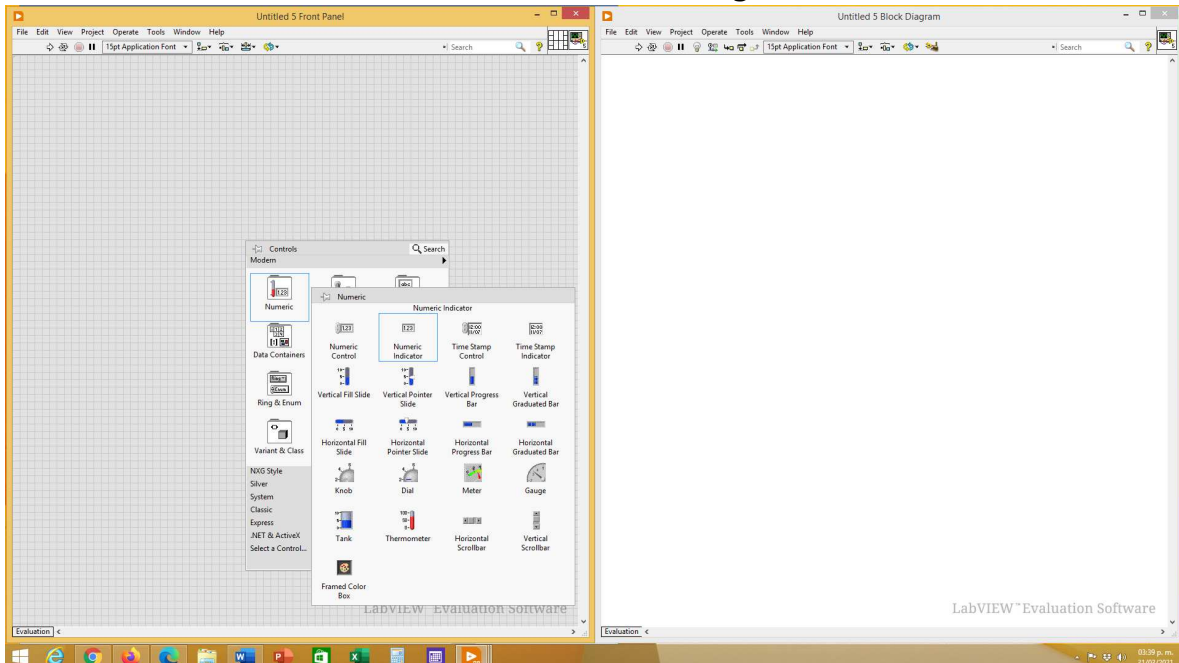
Y aquí se muestra cuando los dos números son diferentes



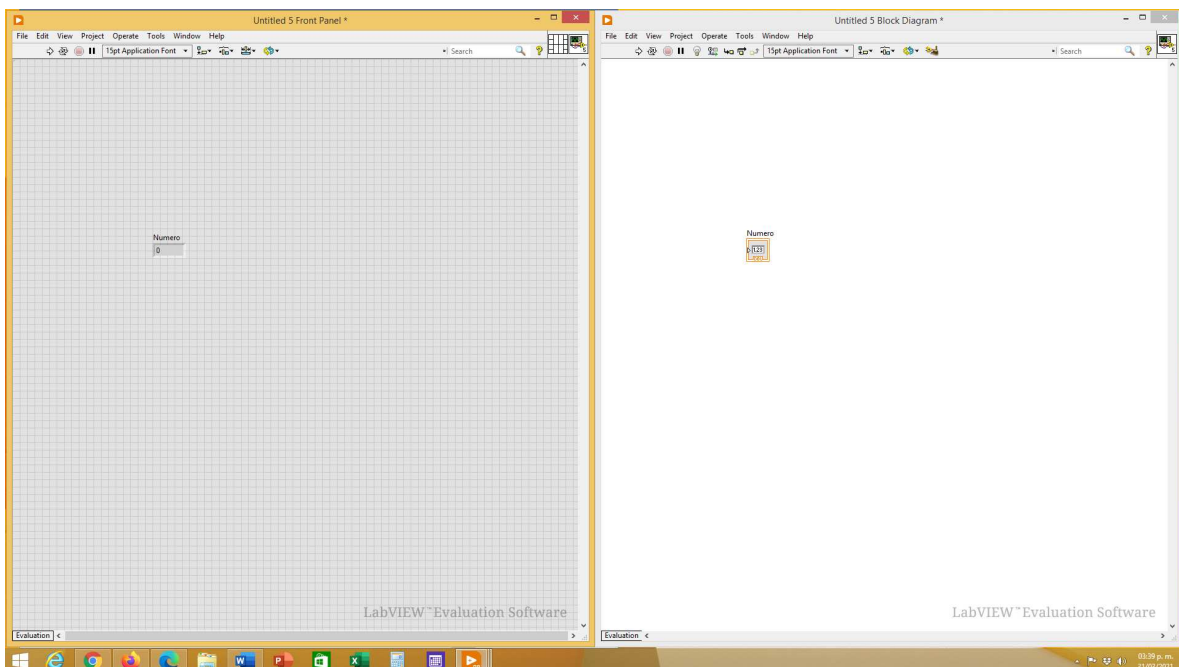
PRACTICA 7 : TIPOS DE DATOS

Abrimos un proyecto nuevo dando clic en Blank VI

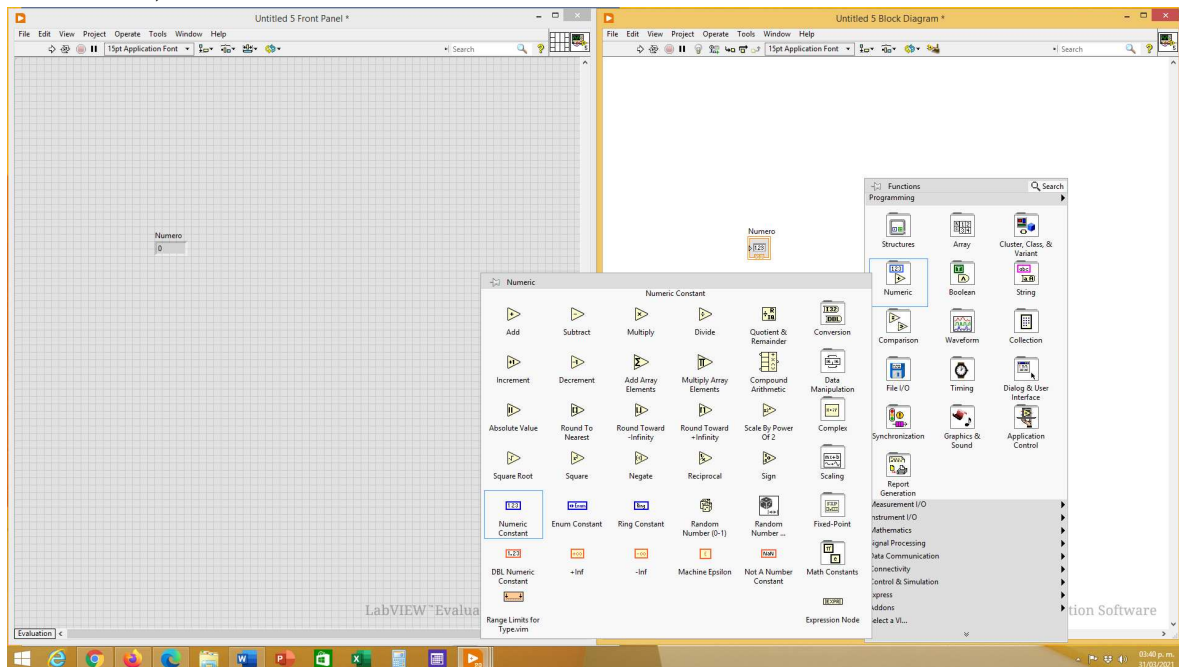
Despues clic con boton derecho para agregar en el Menu Numeric, un indicador Numerico, esto se muestra en la imagen



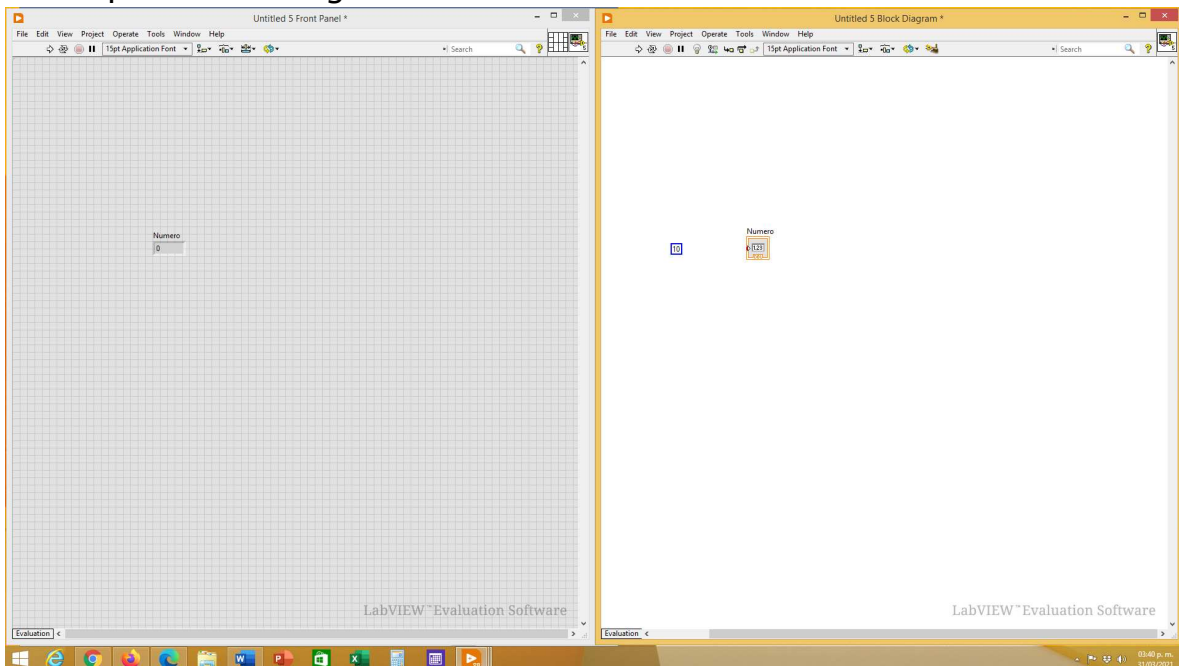
El control genera a su vez un bloque que se muestra en la ventana de la derecha



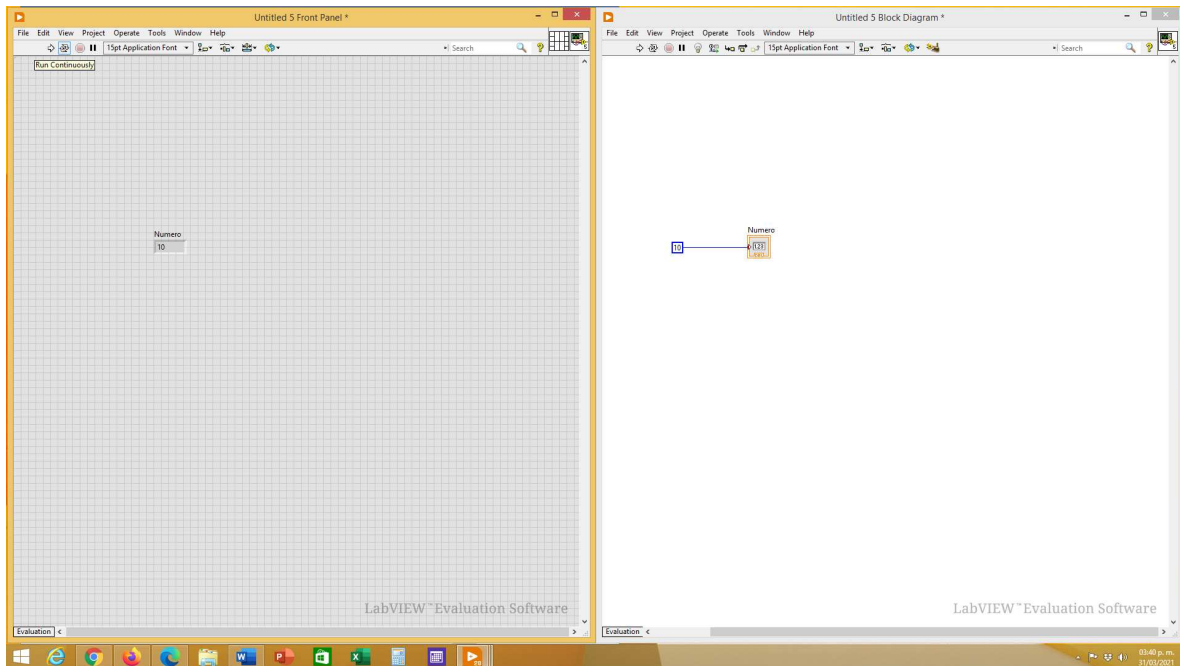
En la ventana de bloques, ingresamos un Numeric Constant (Constante Numerica)



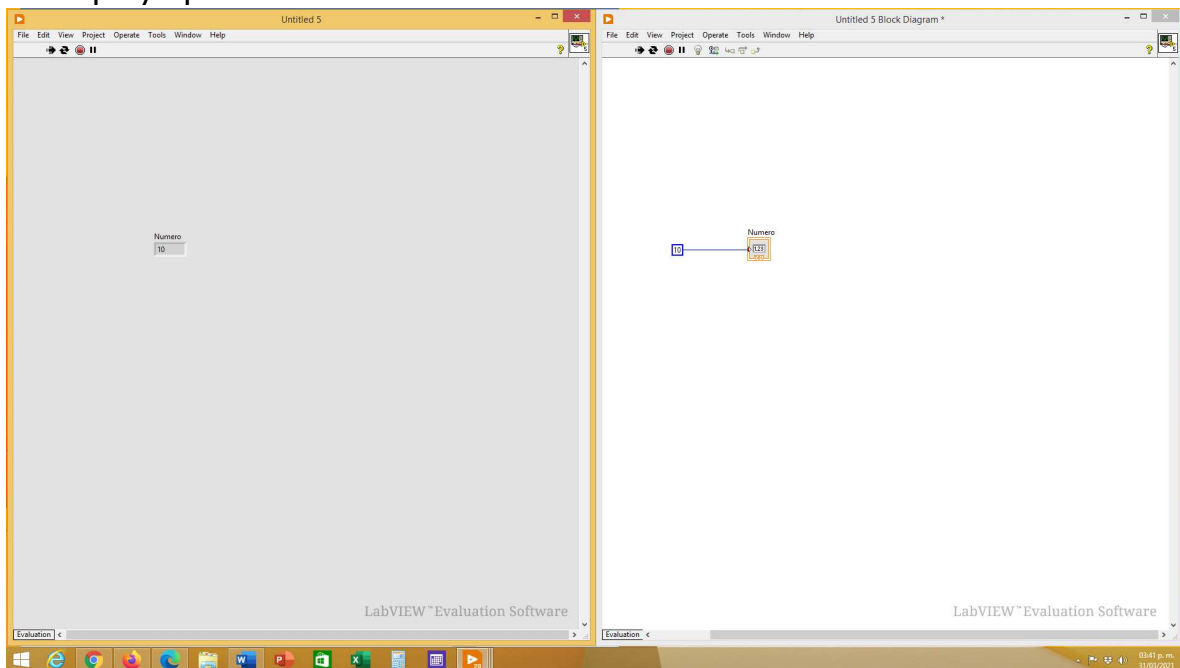
Debe quedar de la siguiente manera:



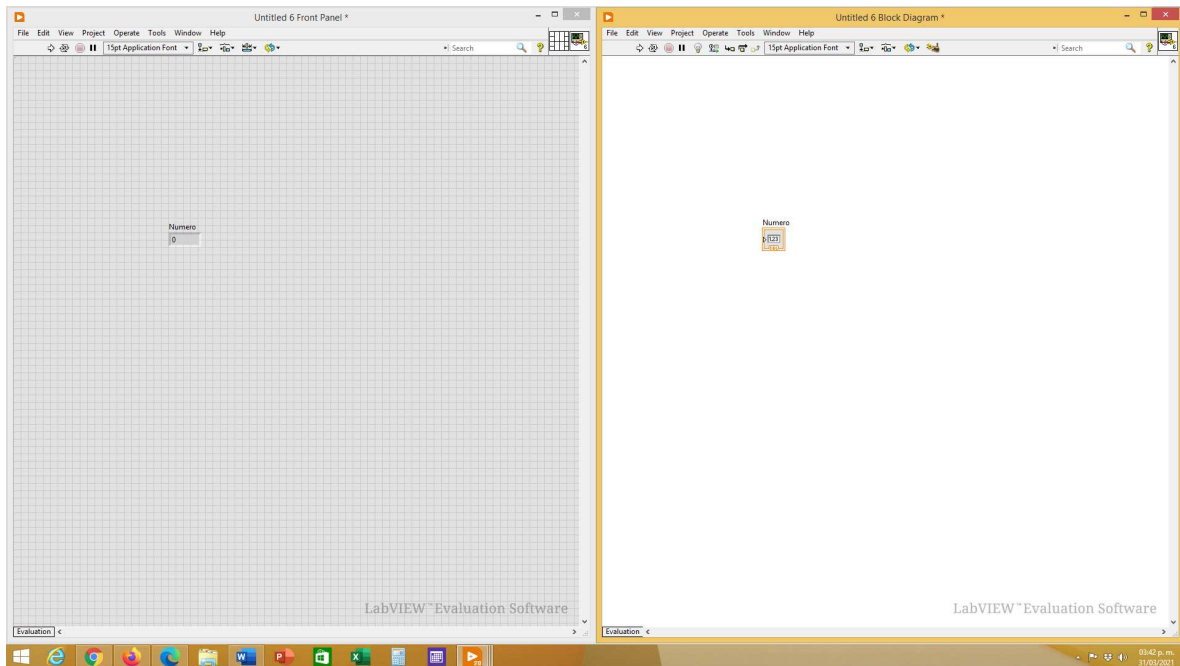
Realizamos la conexión como se muestra tomando como entrada el numero constante y como salida el control numerico (display)



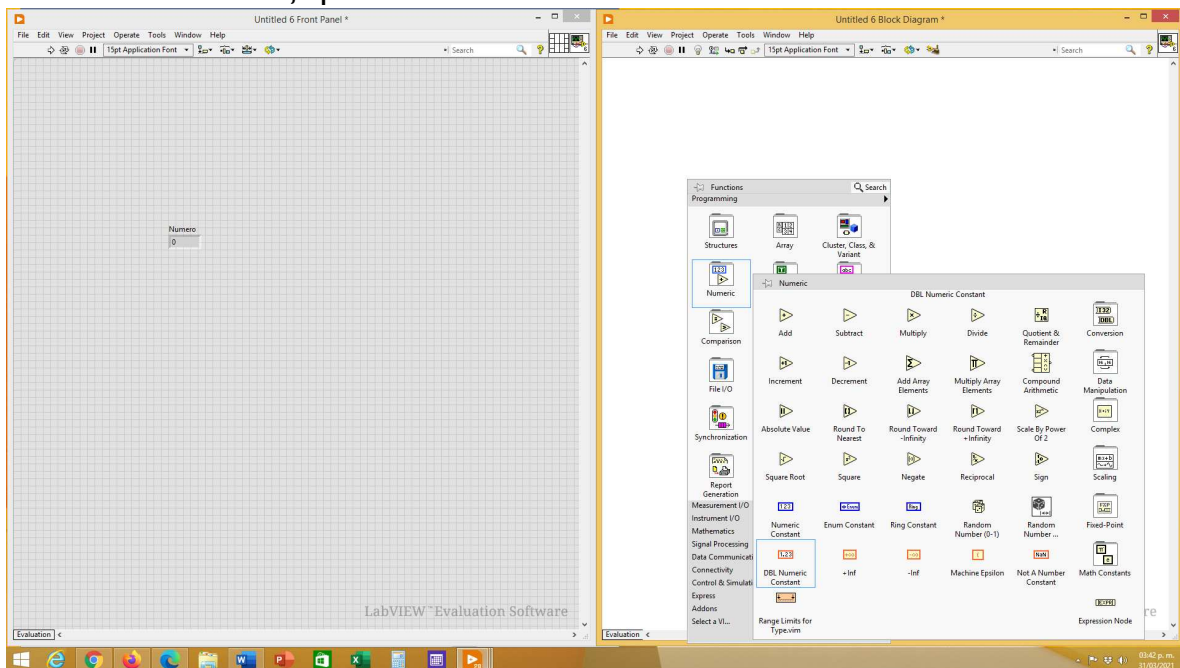
Si le colocamos el valor de 10 en la constante, al ejecutar la aplicación, en el display aparecerá el número 10.



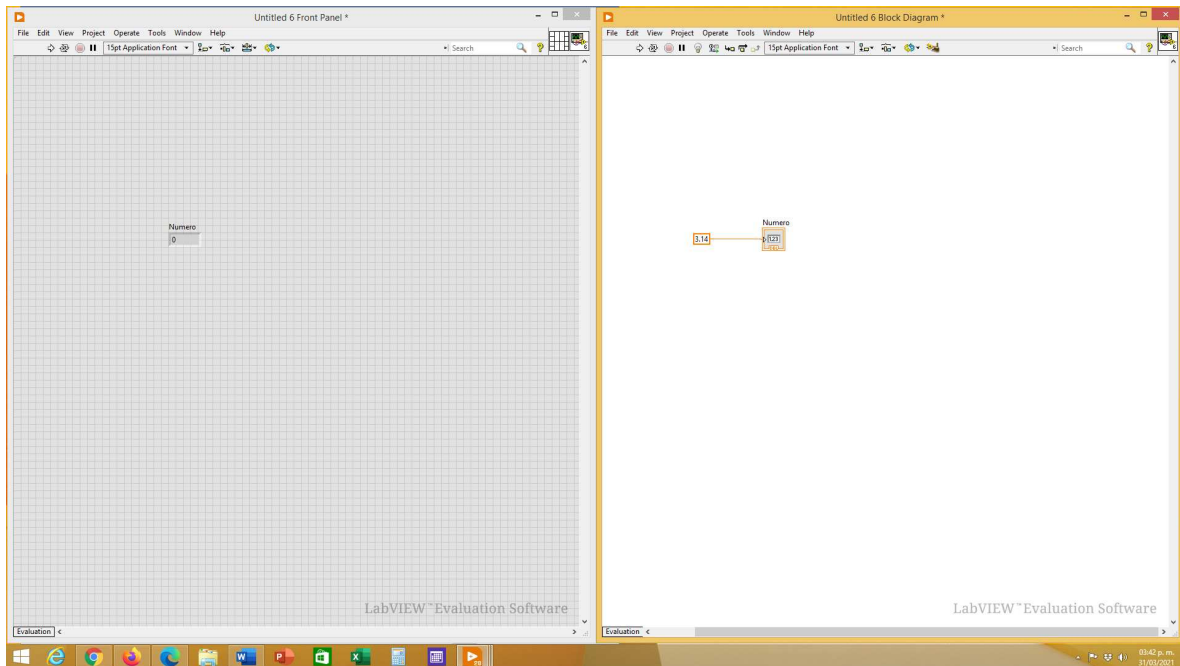
Ahora, vamos a ingresar un Indicador numérico para mostrar un número decimal.
Igual que el procedimiento anterior, agregamos dicho control



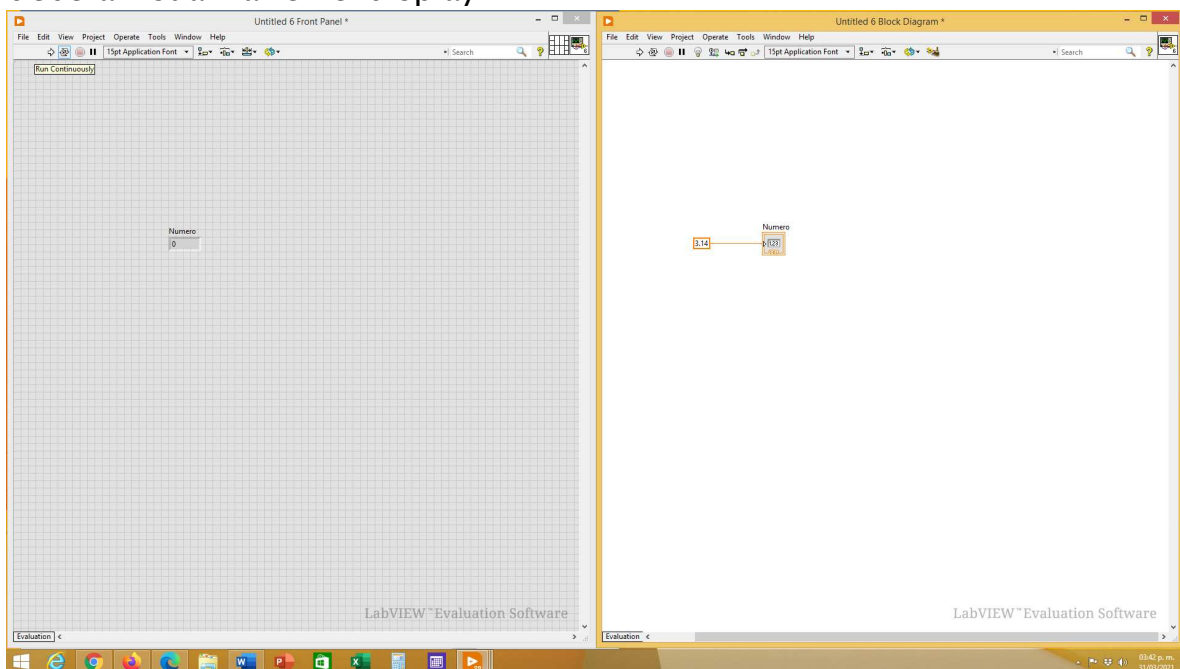
En la parte de los bloques en la sección de Numeric, buscamos DBL Numeric Constant, que es una constante Decimal



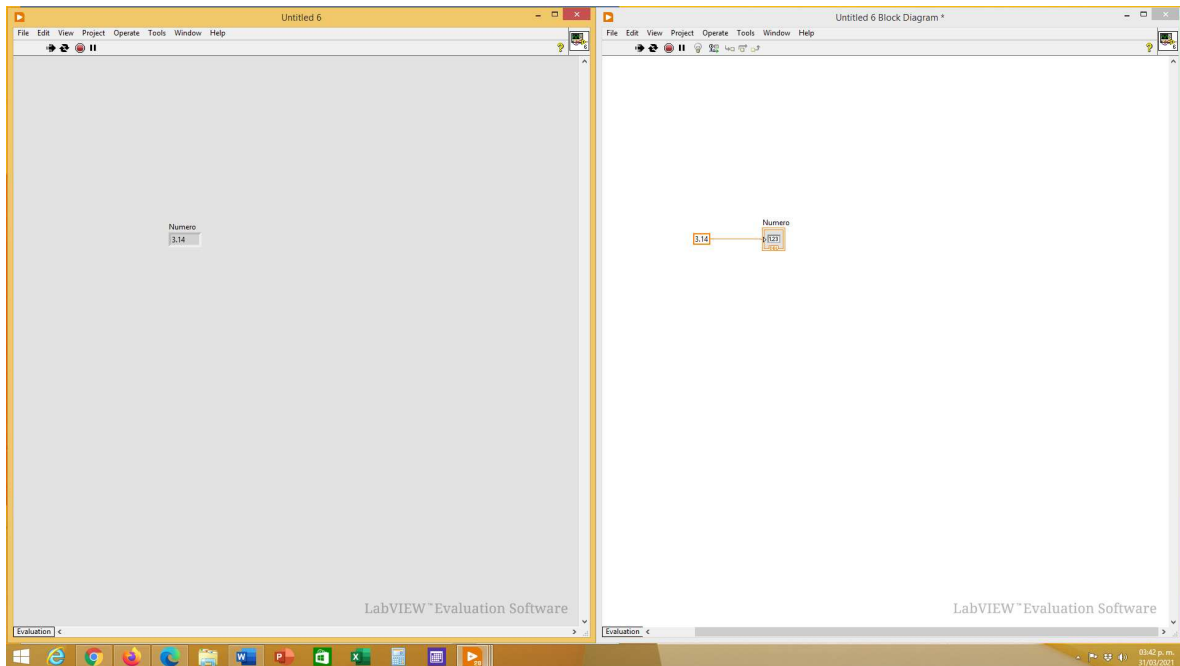
Conectamos ambos bloques de tal forma que la constante apunte a el control numerico como se muestra en la imagen.



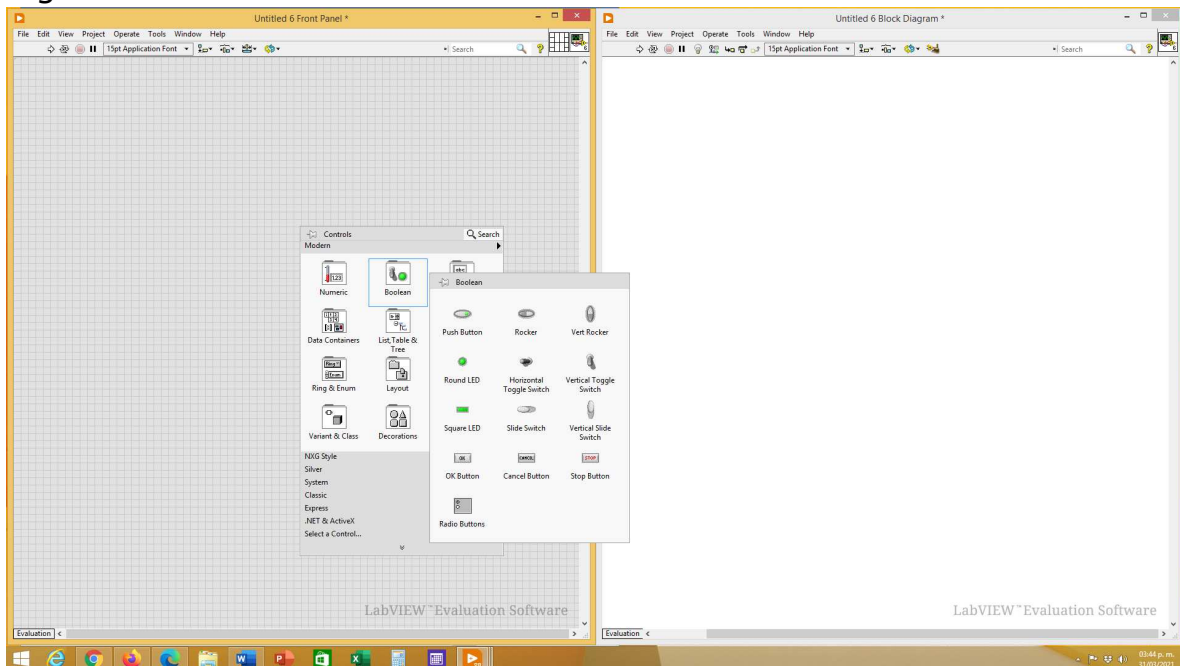
Ejecutamos dando clic en las flechas blancas en forma de círculo y colocamos un numero con decimales en la constante. Este numero se debera visualizar en el display



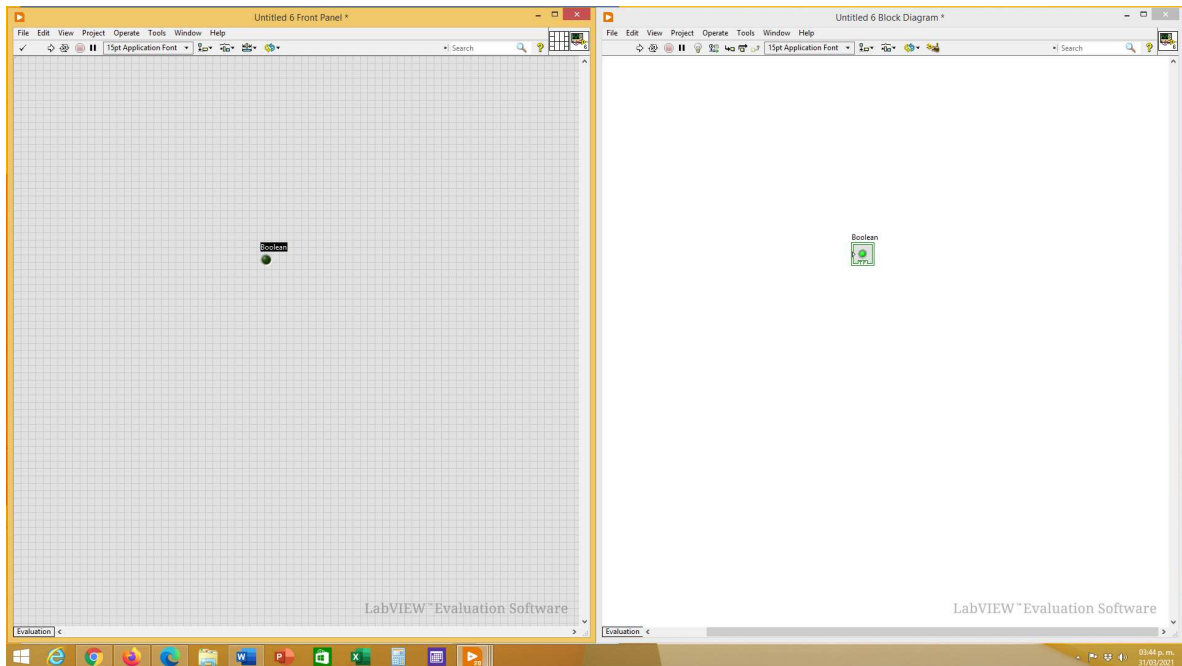
Este seria el resultado:



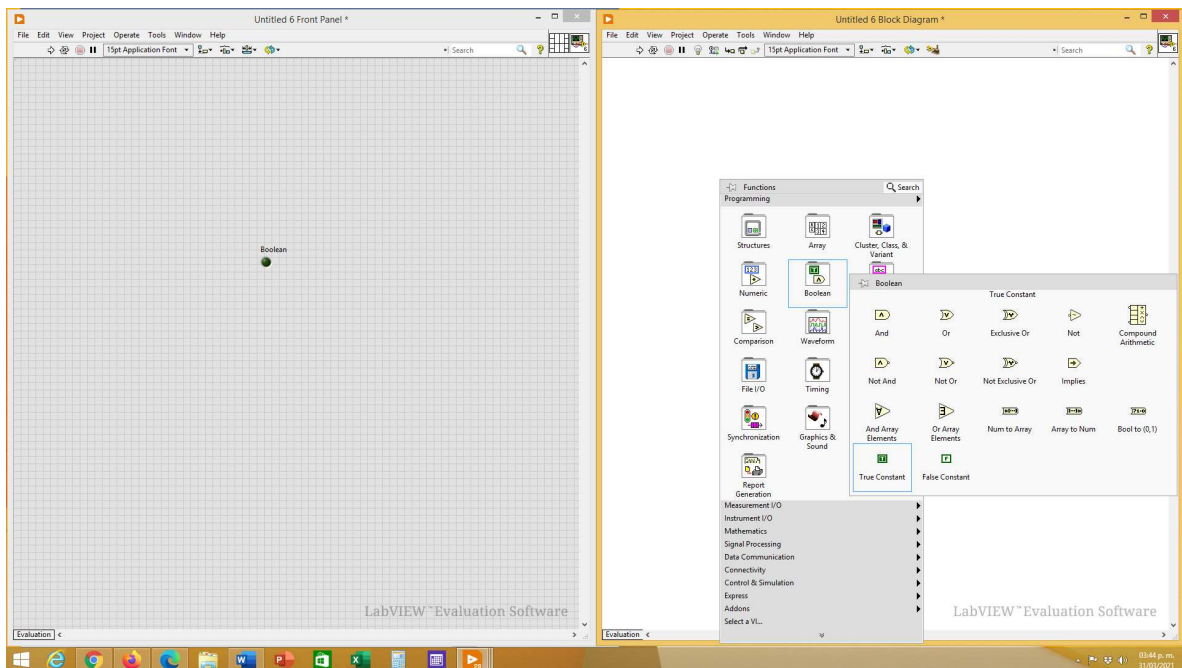
Ahora probaremos con variables Booleanas:
Ingresaremos un led en el menu de controles Boolean



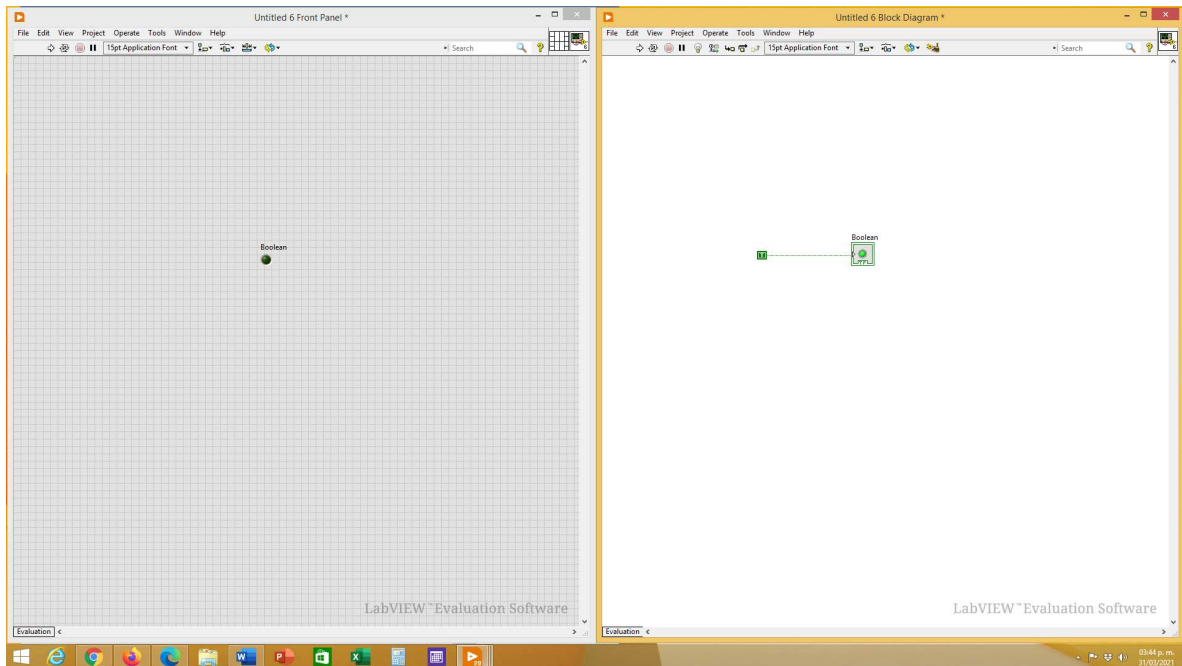
Debe quedar de esta manera. El nombre que aparece arriba se puede cambiar dando doble clic sobre el texto



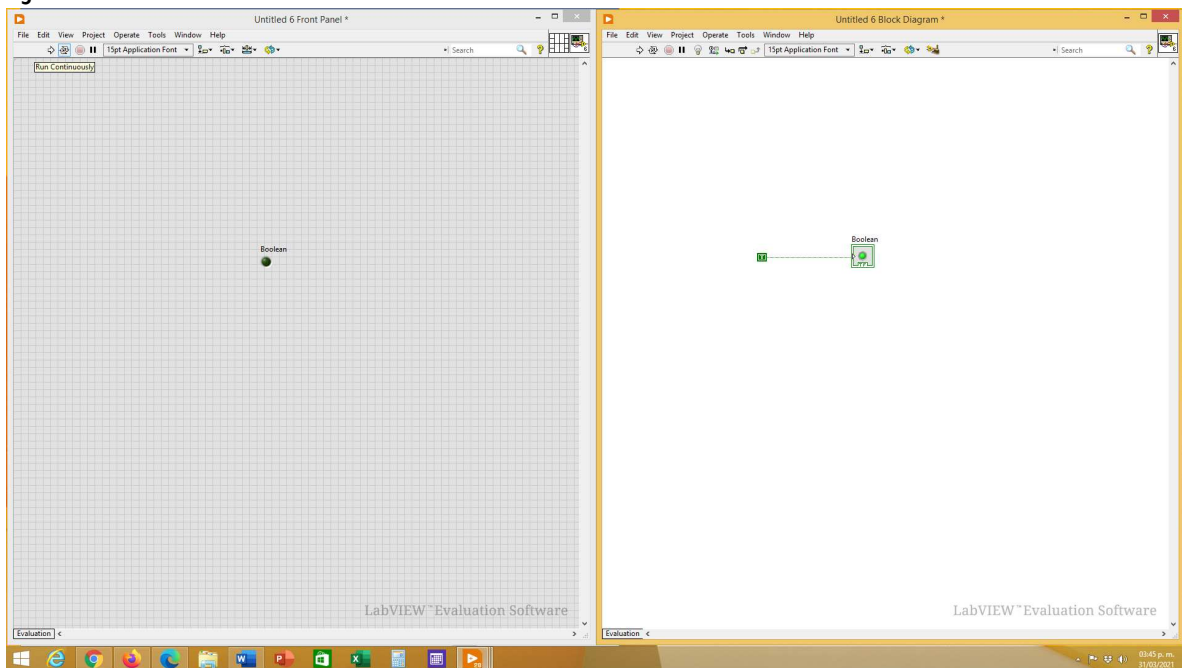
Despues en la ventana de bloques, damos clic en Boolean y seleccionamos True Constant



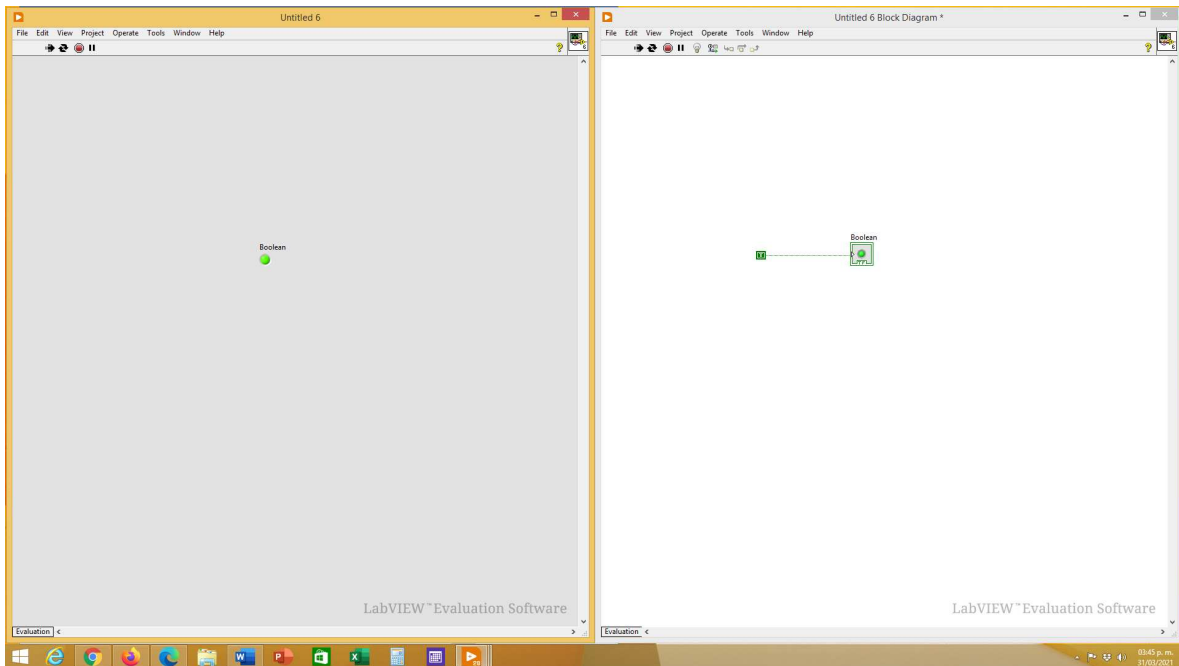
Lo ingresamos y conectamos la variable con el bloque del led



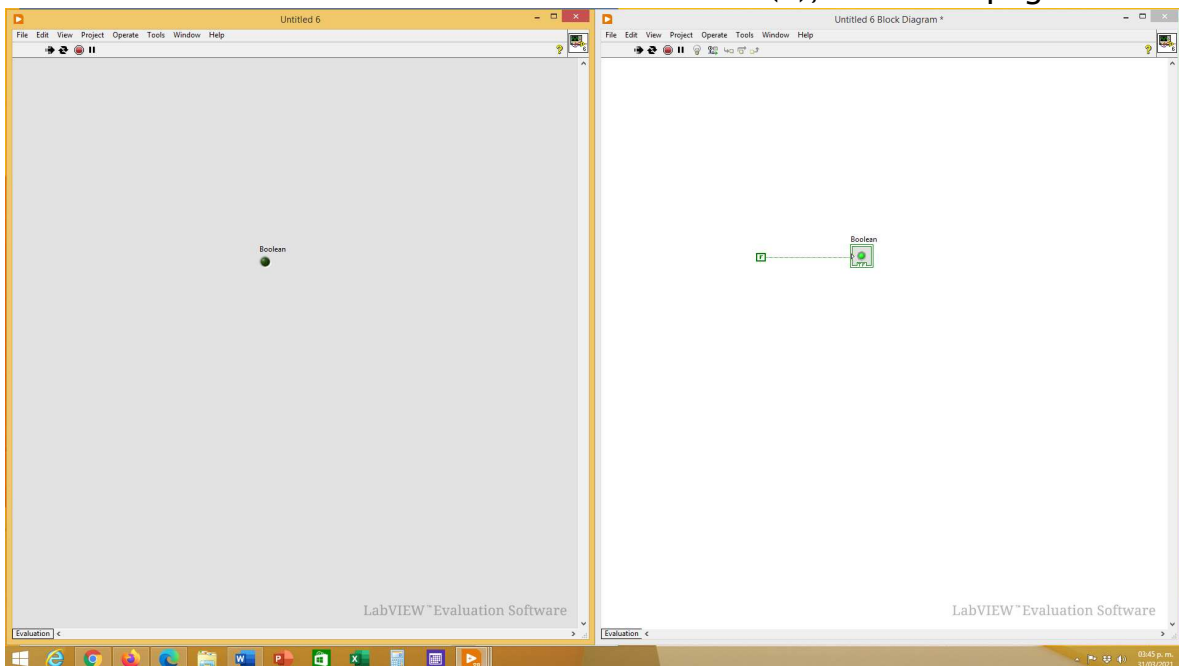
Ejecutamos dando clic en las flechas blancas



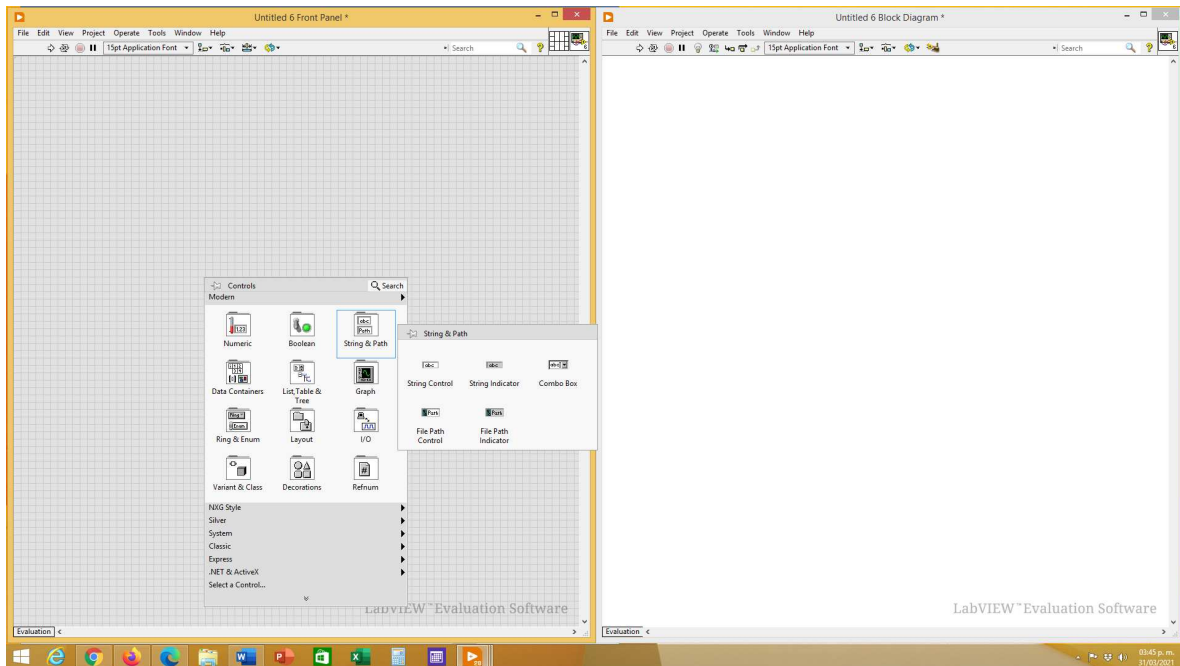
Como se le mando un valor TRUE (1), entonces el led debe encender al ejecutar la aplicación.



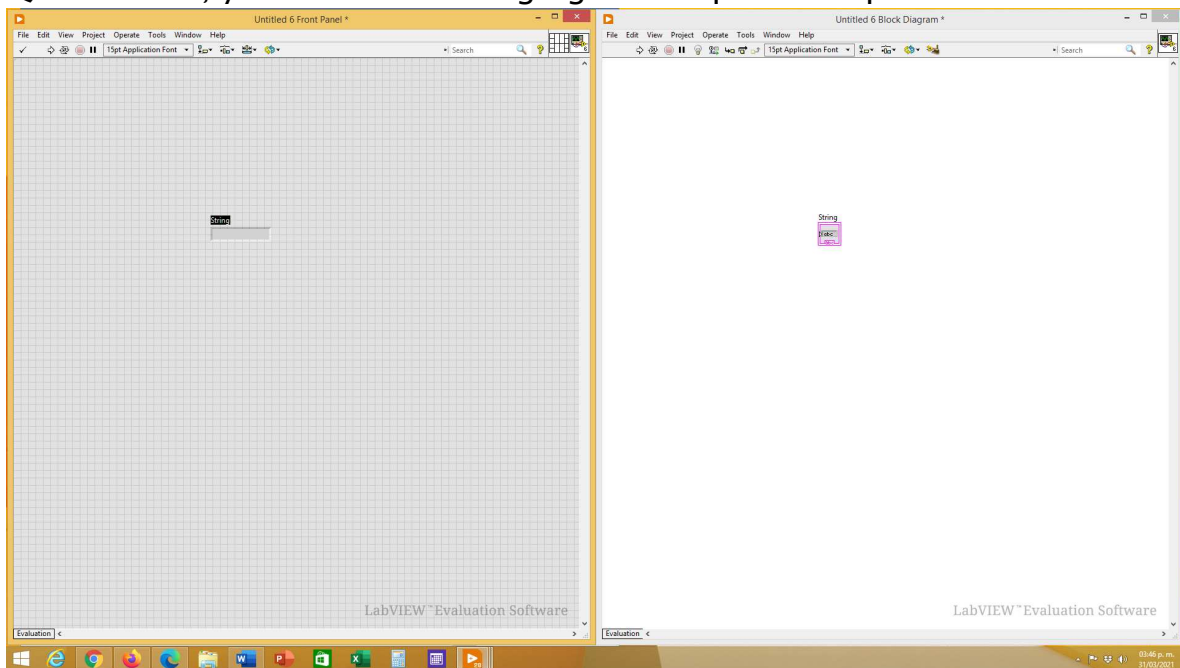
En caso de usar un valor de FALSO en la variable (0), el led se apagará



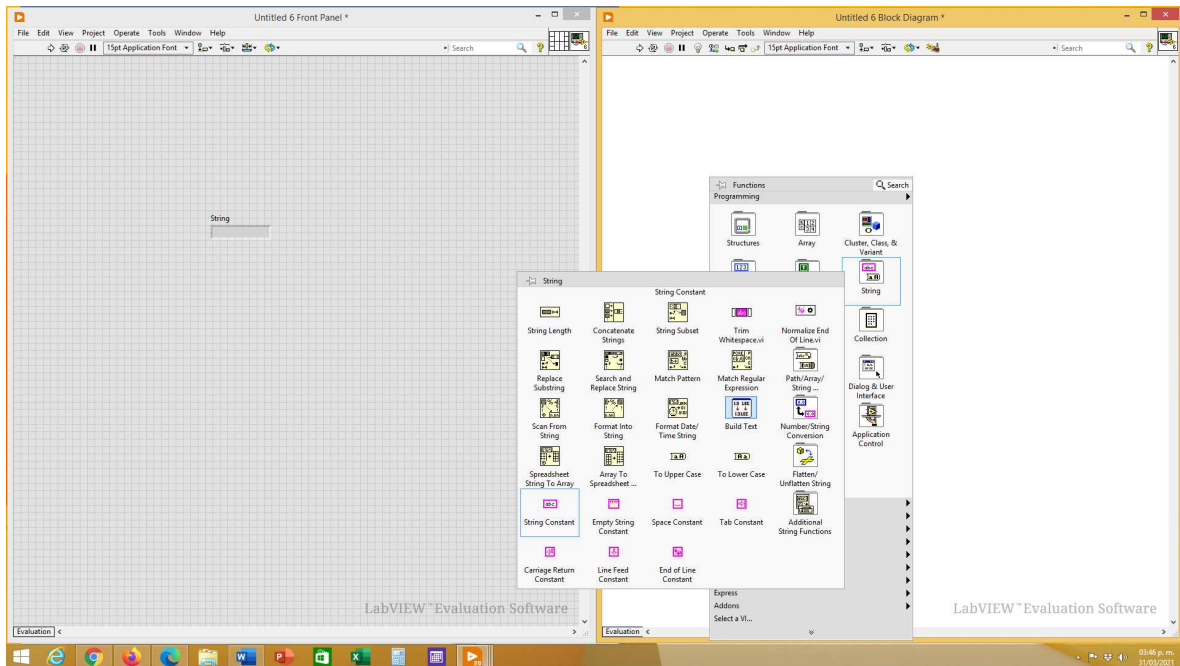
Otro tipo de datos que podemos manejar, es de de Cadena o String. Seleccionamos un control tipo cadena (String Indicator) dando clic secundario en la ventana de Panel frontal.



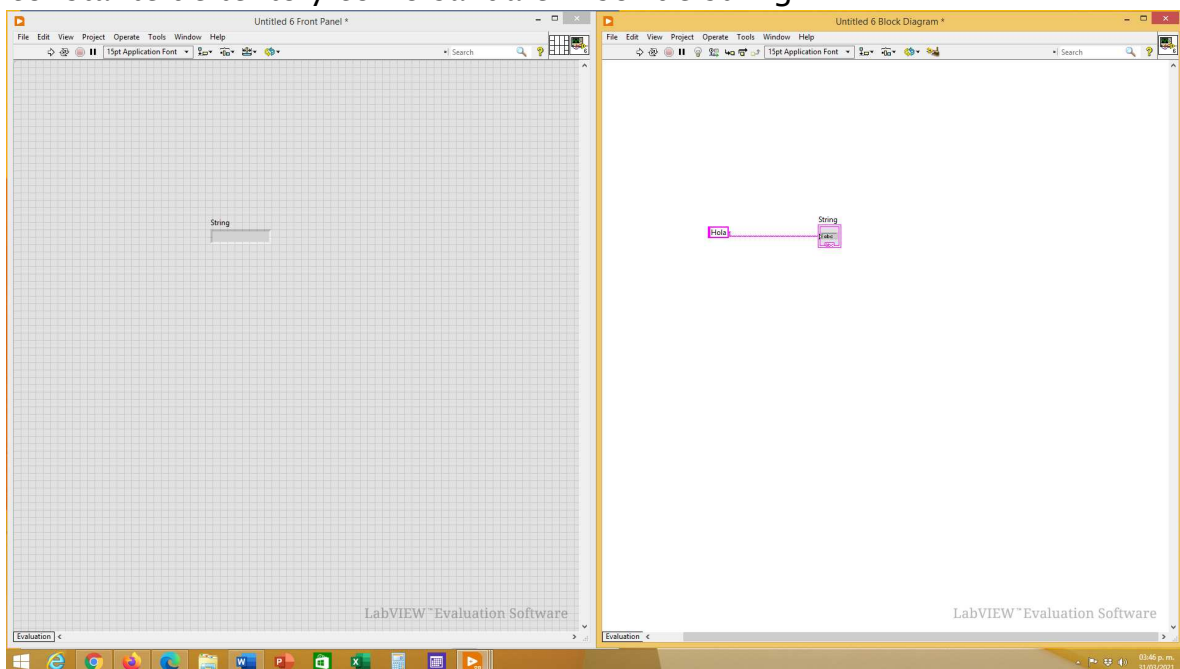
Quedaria asi, y vemos como se agrega el bloque correspondiente



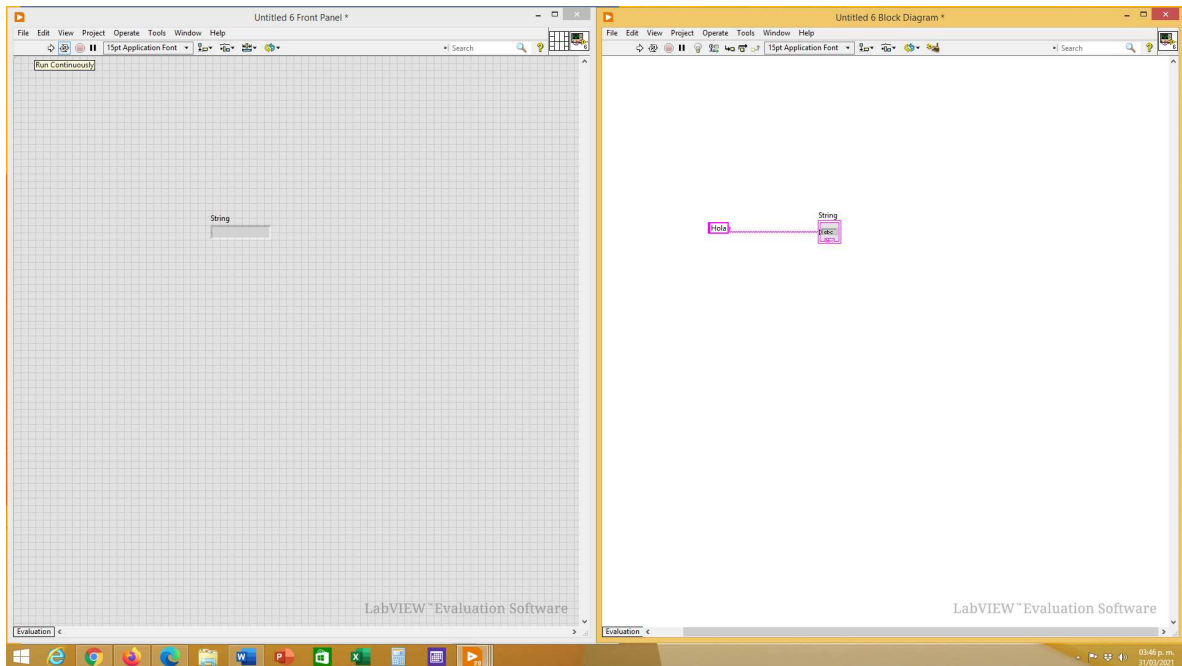
En la ventana de los bloques, Seleccionamos String y a continuación String Constant



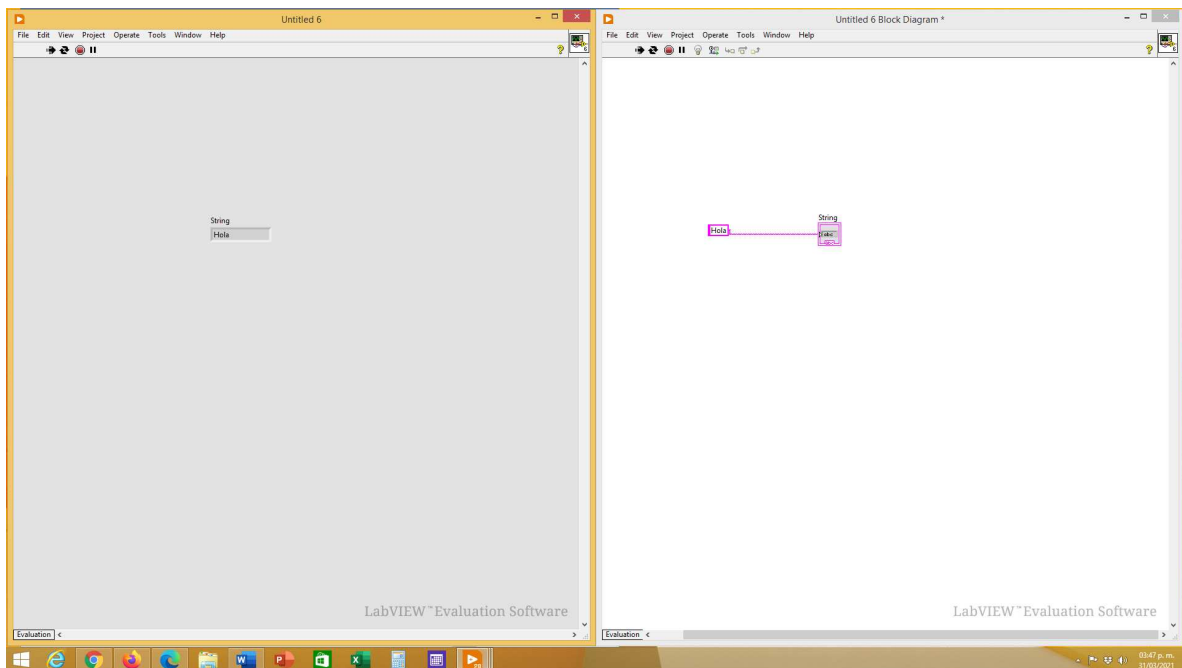
Esa variable debe llevar un valor de texto. Por ejemplo, podemos escribir Hola. Despues realizamos la conexión, tomando como entrada la constante de texto y como salida el visor de string



Ejecutamos la aplicación dando clic en las flechas blancas



En el control de String aparecera el texto que asignamos previamente en la variable.

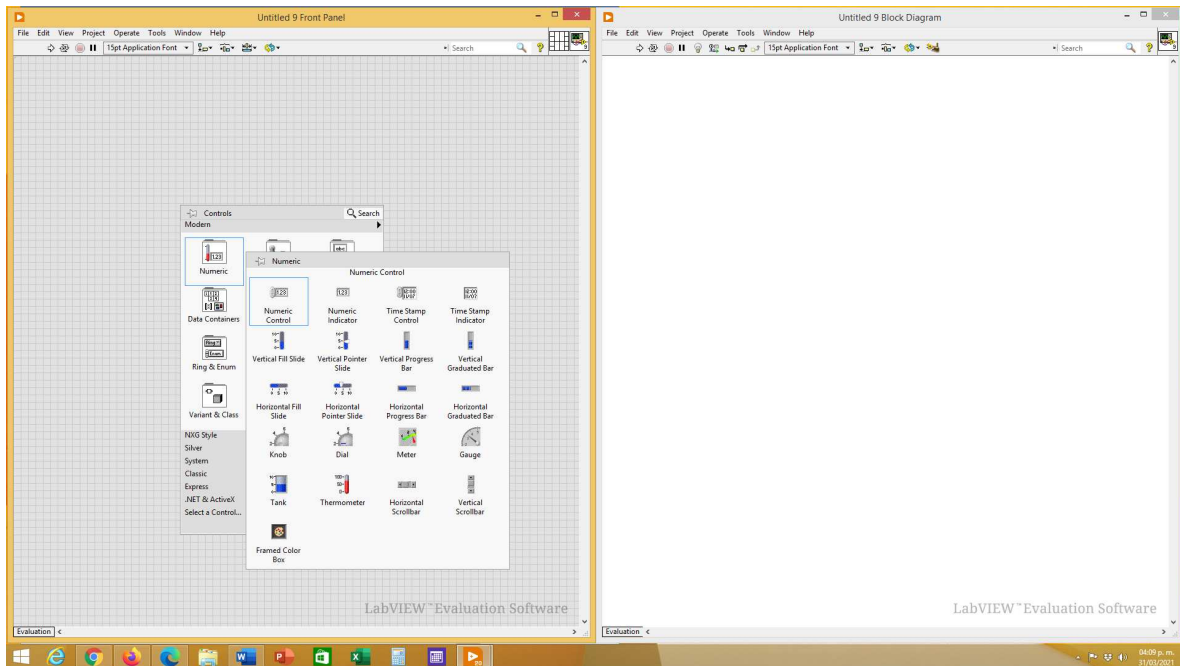


PRACTICA 8: SUMA DE DOS NUMEROS USANDO VARIABLES.

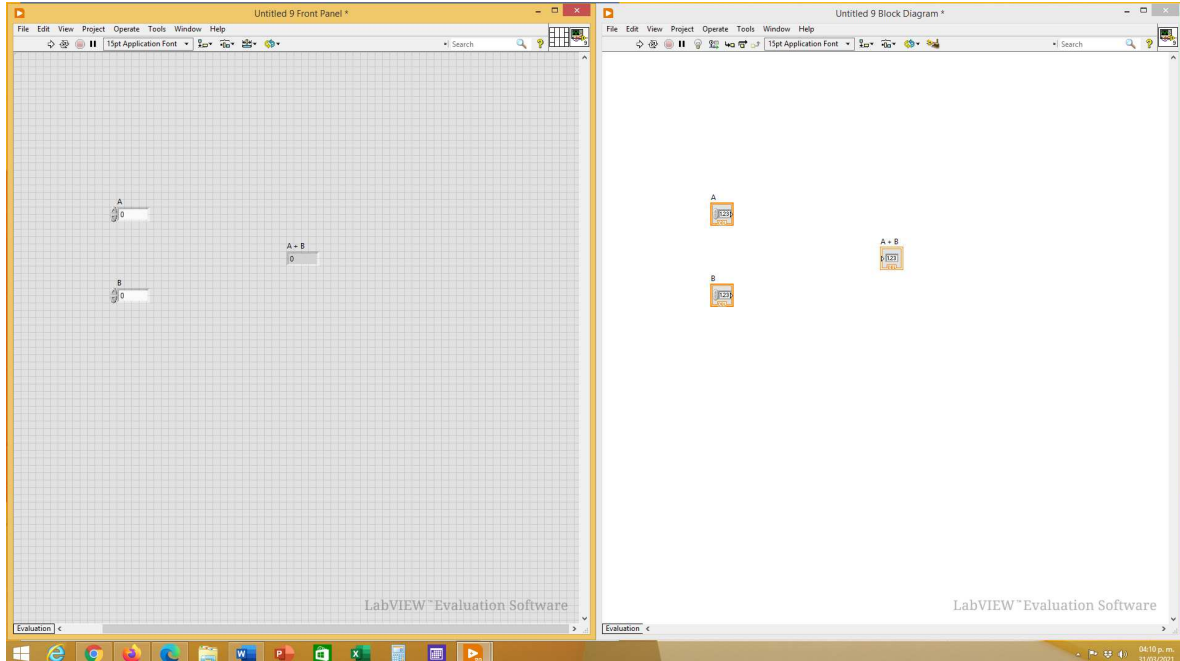
Recordando el ejercicio de sumar dos numeros, realizaremos la misma suma, solo que ahora, el dato que se ingrese en cada control, se guardara en una variable diferente para cada control. Despues estas variables seran sumadas y el resultado almacenado en una variable de salida que sera mostrada en un visor tipo display

Abrimos un proyecto nuevo dando clic en Blank VI

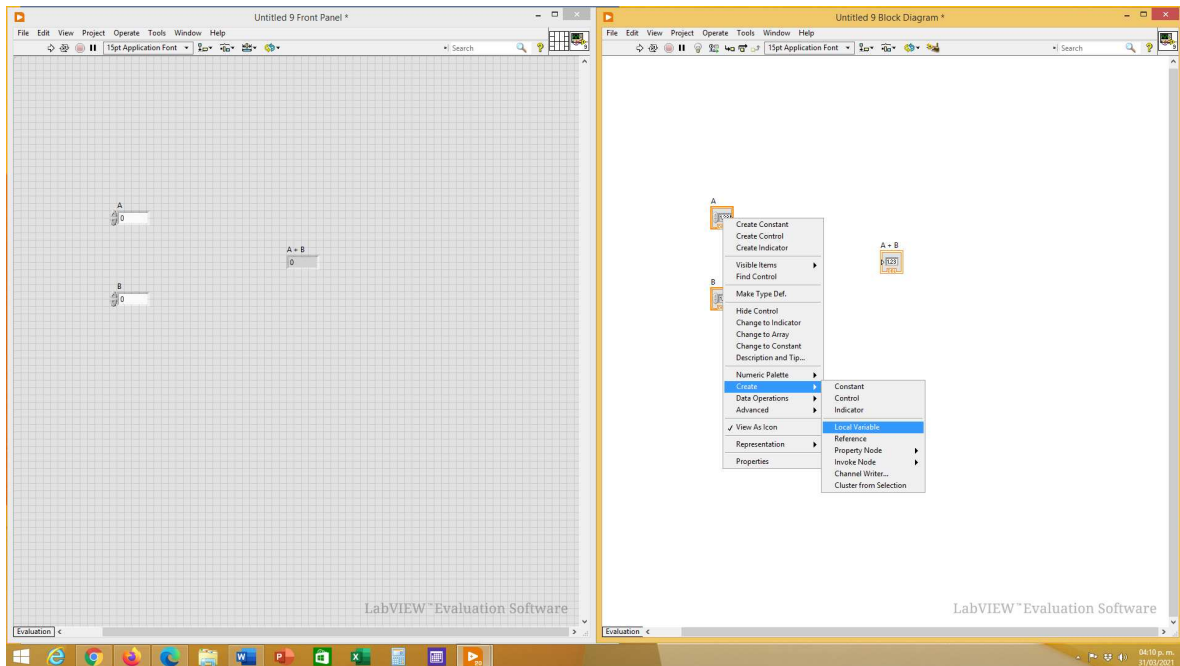
Despues clic con boton derecho para agregar en el Menu Numeric, un control Numerico, esto se muestra en la imagen



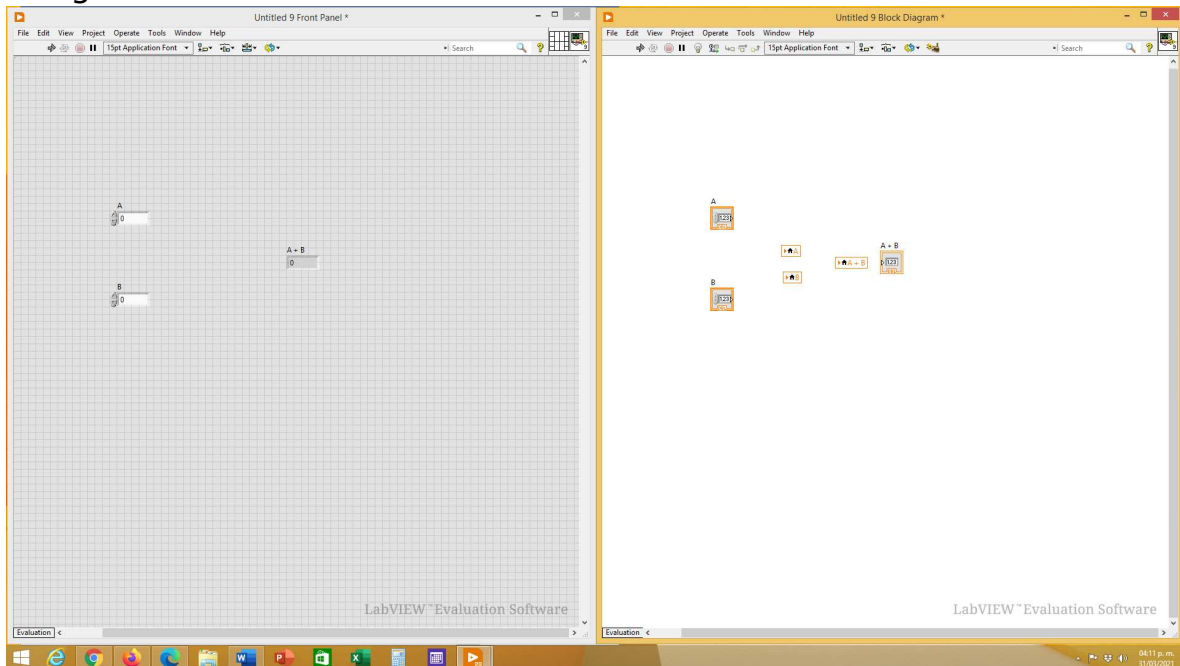
Ingresamos dos controles y les ponemos como etiquetas A y B. Tambien ingresamos un visor numerico cuya etiqueta sera A+B



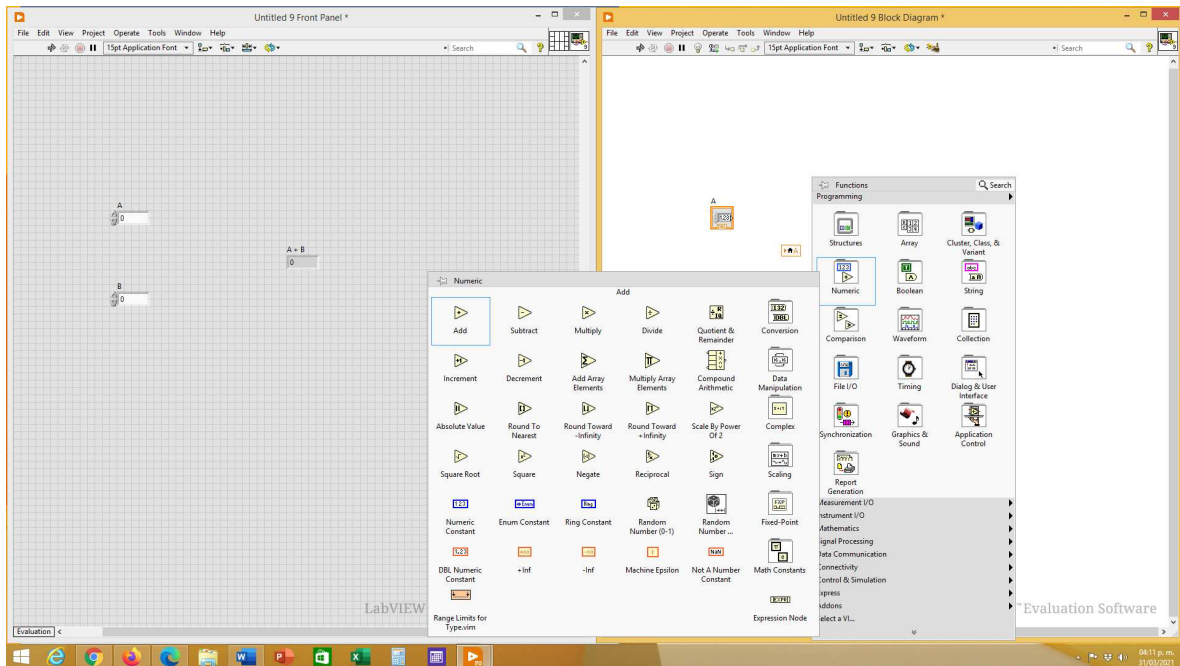
Del lado de los bloques, damos clic con boton derecho sobre el bloque de A y elegimos Create Local Variable



Hacemos lo mismo para la otra entrada y para la salida. Esto quedara de la siguiente manera:

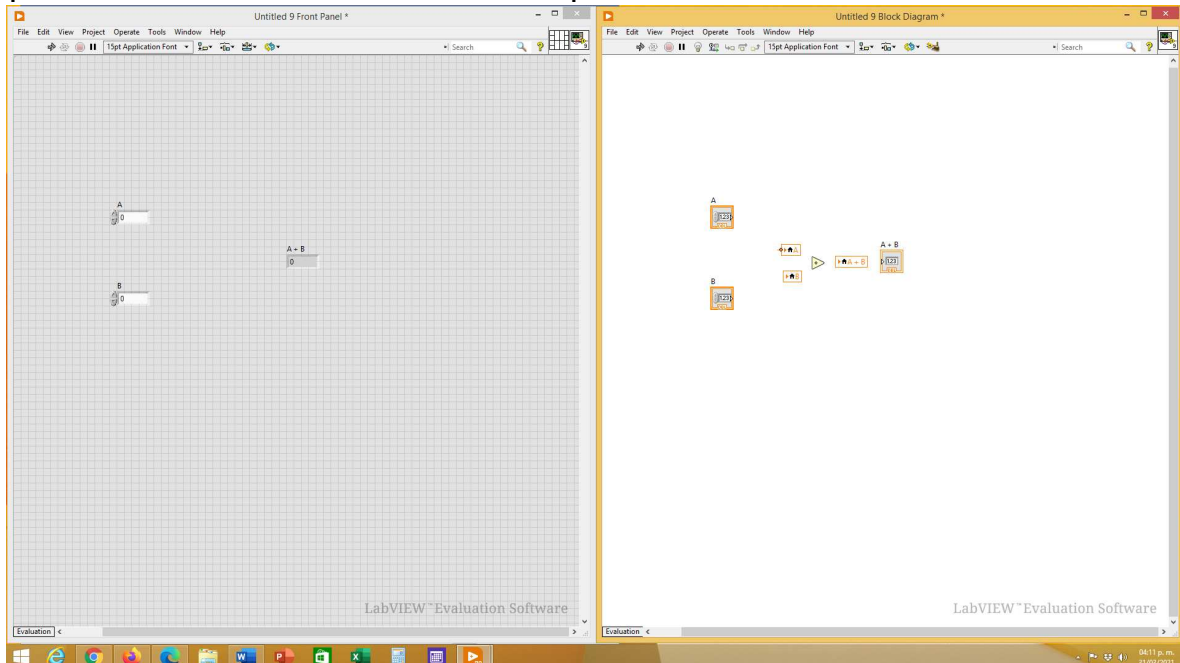


Despues metemos el contro Numerico Add para realizar la suma

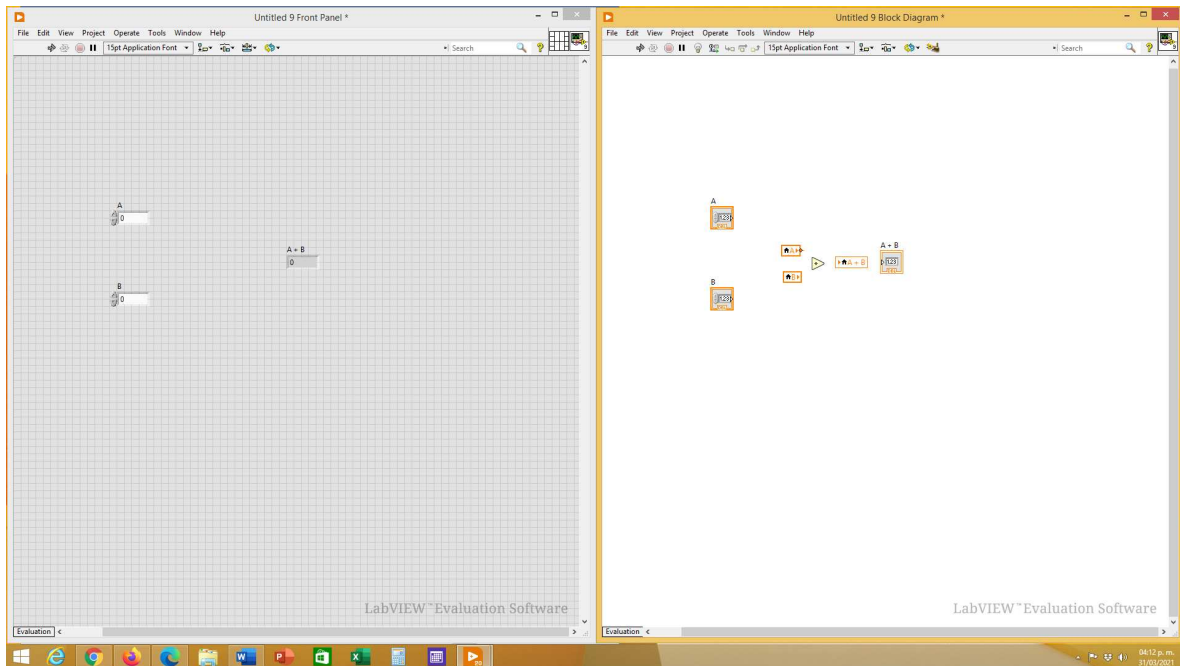


En este caso lo que conectaremos serán las variables directamente al control Add.

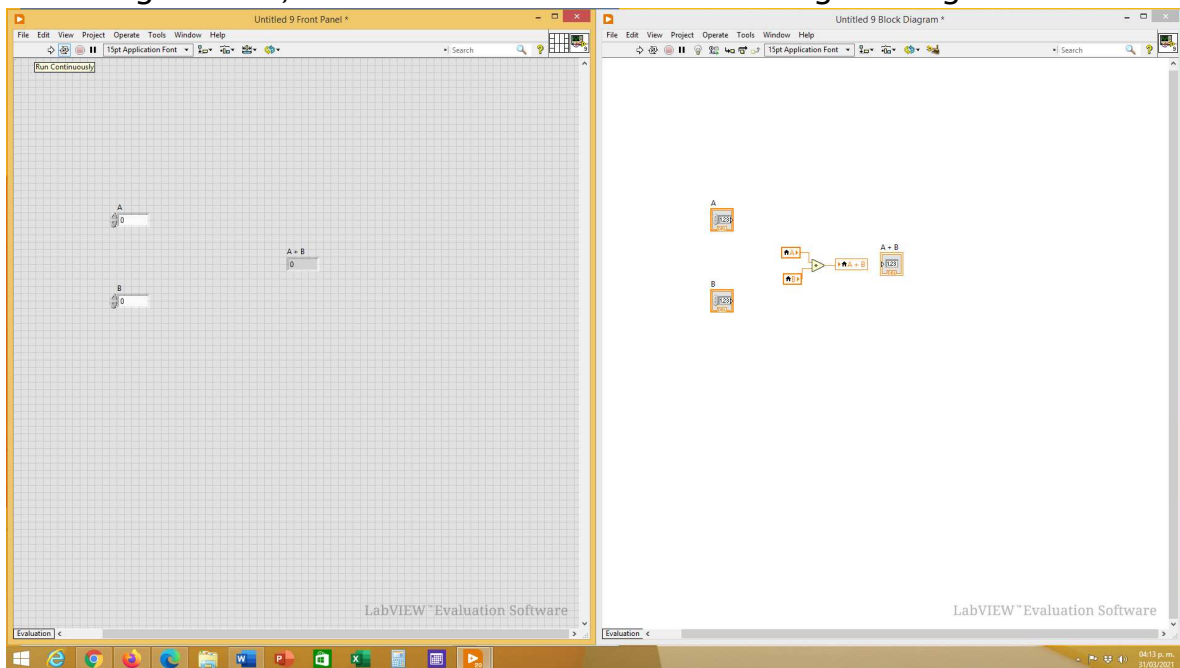
Observaremos que no nos permitira conectar las variables de entrada, pues sus salidas estaran del lado izquierdo



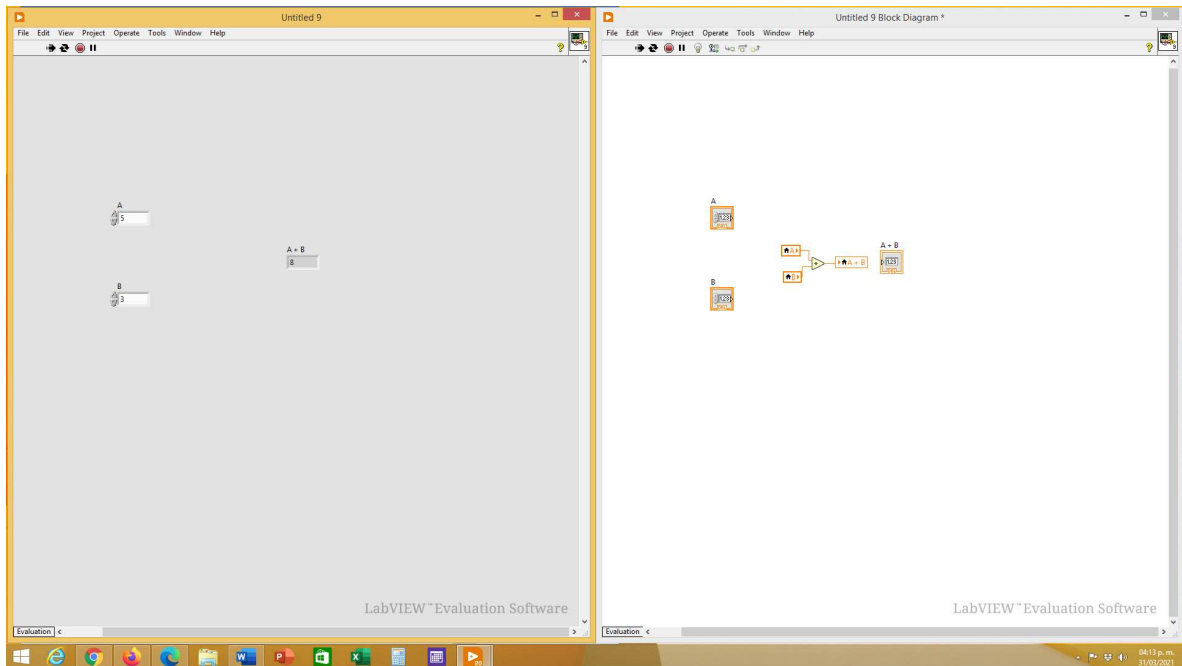
Corregimos esto dando clic con boton derecho en la variable y cambiamos a Write en vez de Read, esto lo hacemos en las dos variables de entrada.



Ya corregido esto, entonces ahora si conectamos según la figura



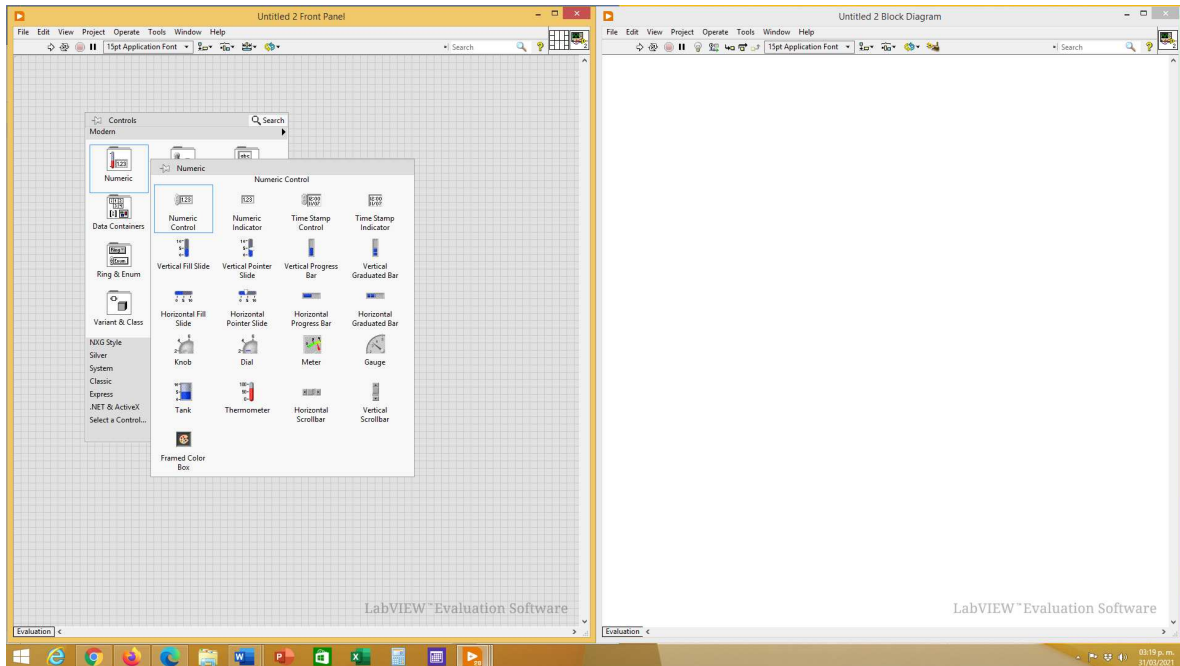
Ejecutamos la aplicación dando clic en las flechas blancas y observaremos el resultado de salida al cambiar los valores de las dos entradas.



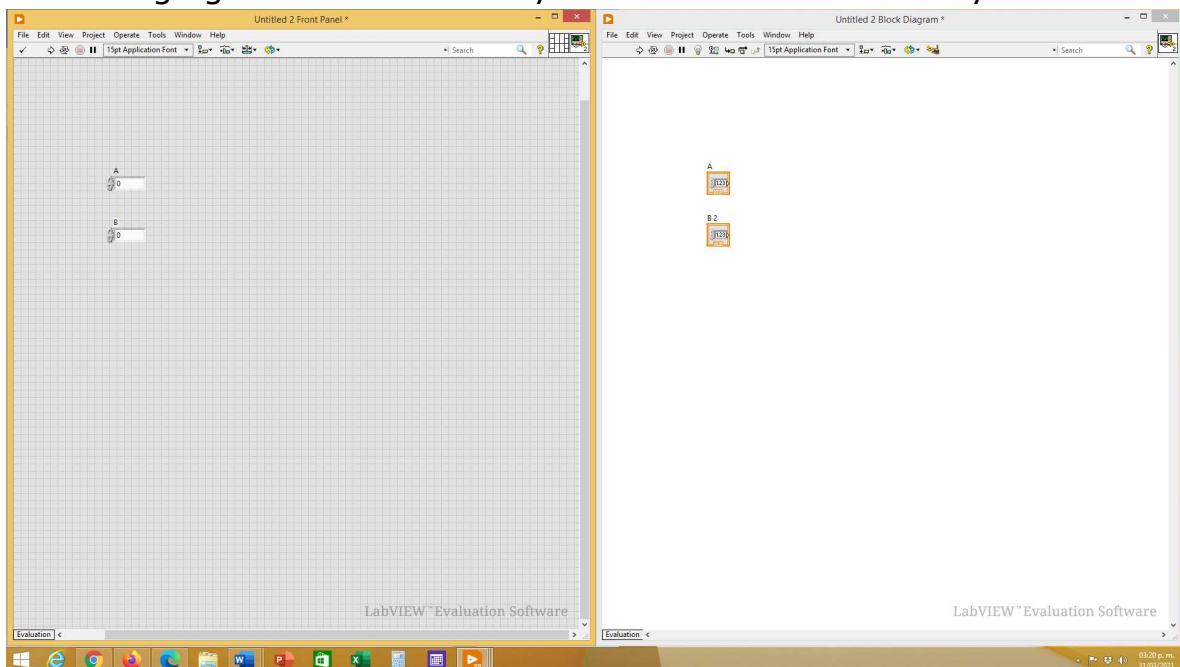
PRACTICA 9: OPERACIONES MATEMATICAS BASICAS ENTRE 2 NUMEROS

Abrimos un proyecto nuevo dando clic en Blank VI

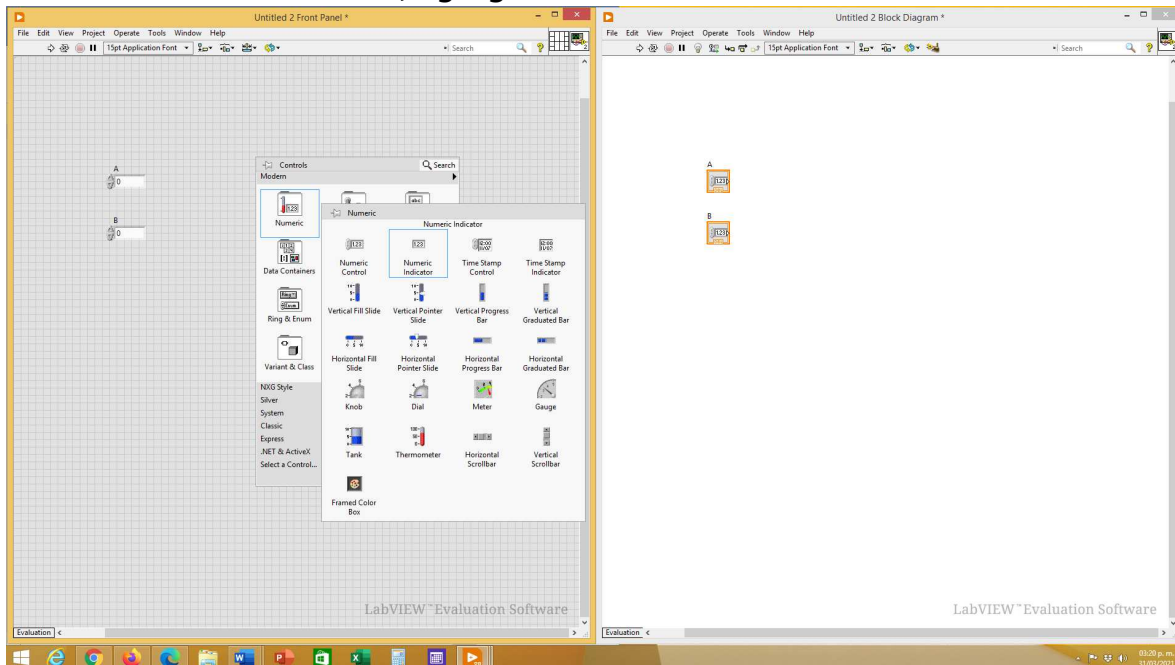
Despues clic con boton derecho para agregar en el Menu Numeric, un control Numerico, esto se muestra en la imagen



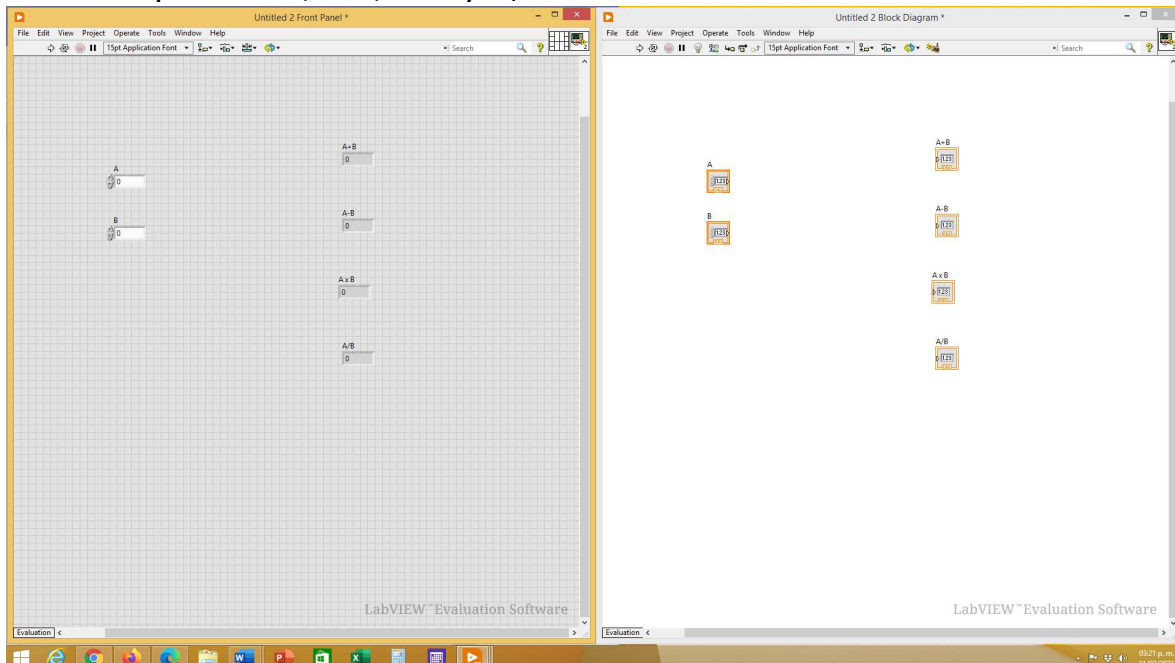
En total agregaremos 2 controles y los nombraremos como A y B



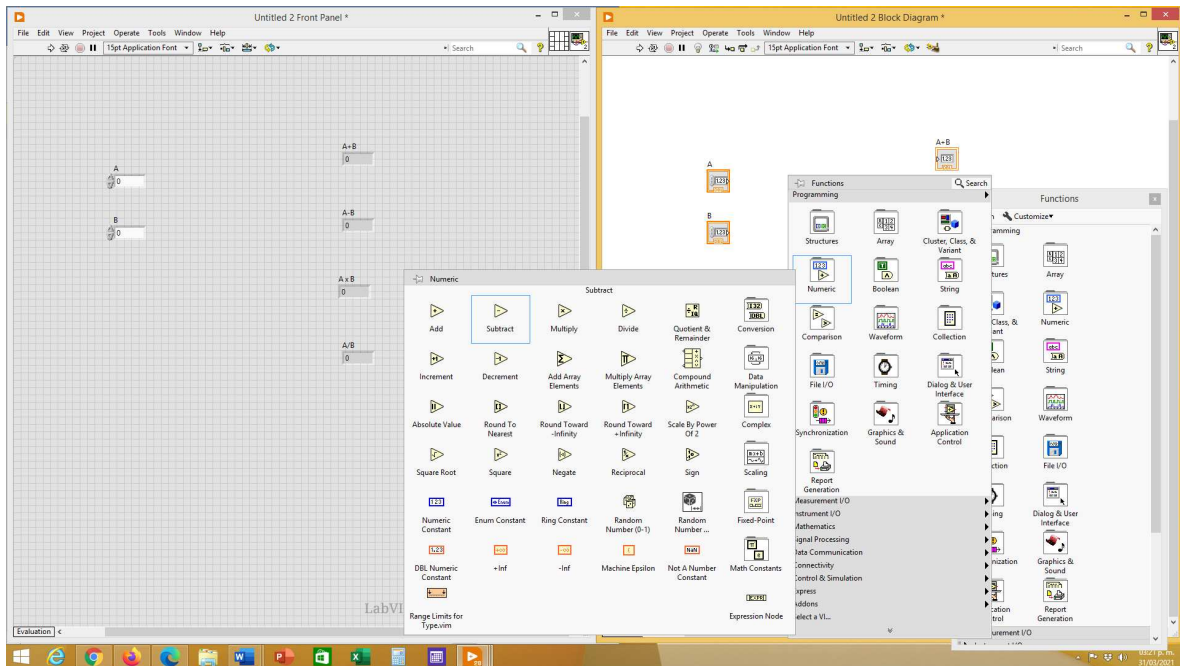
Para mostrar el resultado, agregemos el control Numeric Indicator



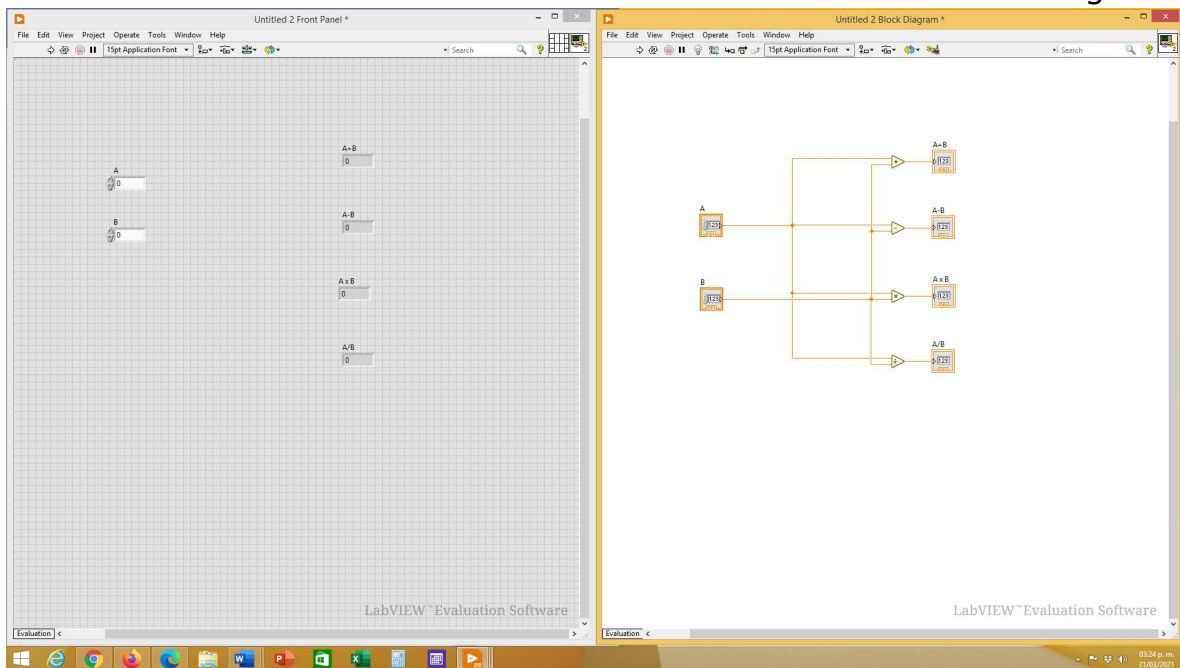
Haremos esto 4 veces y en cada uno de estos 4 controles, pondremos como etiqueta A+B, A-B, AxB y A/B



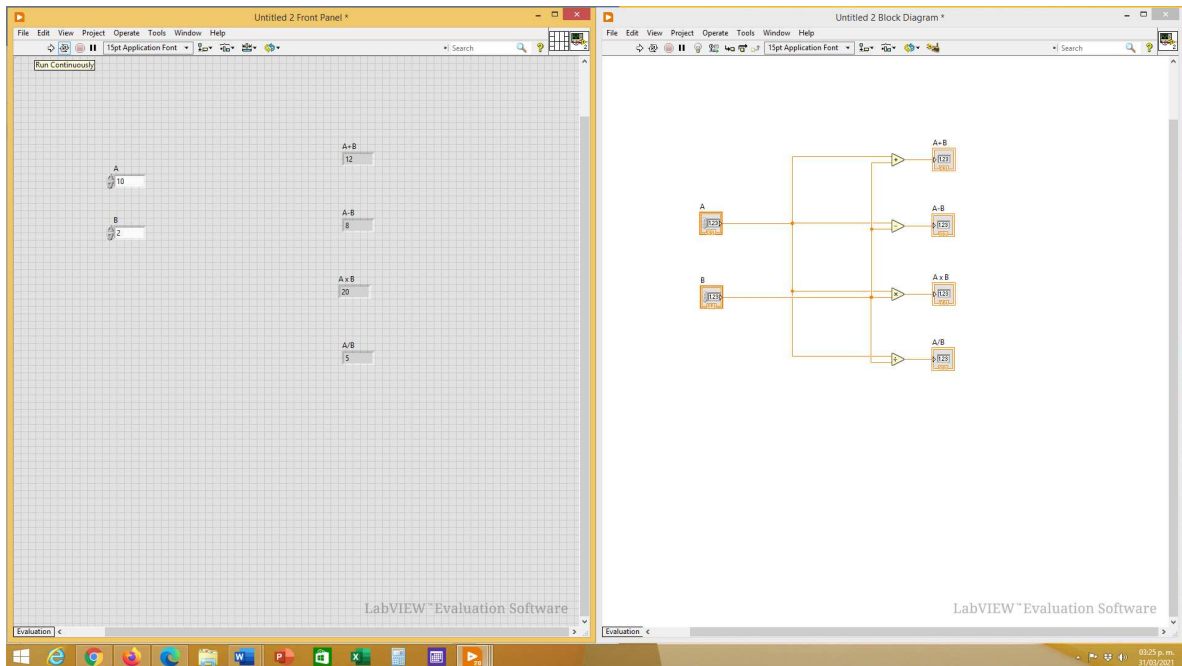
Paso siguiente, consiste en meter los bloques Numericos para la Suma (Add), Resta (Subtract), Multiplicacion (Multiply) y division (Divide)



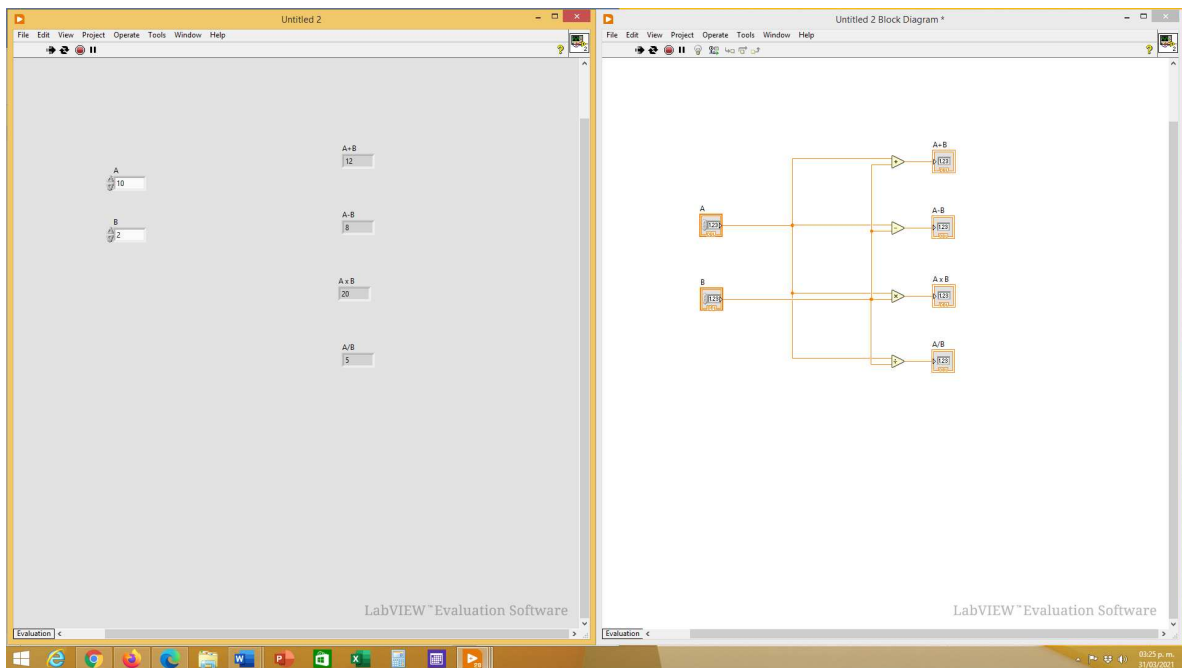
Posteriormente realizamos las conexiones como se observa en la imagen



Para ejecutar, damos clic en las flechas blancas



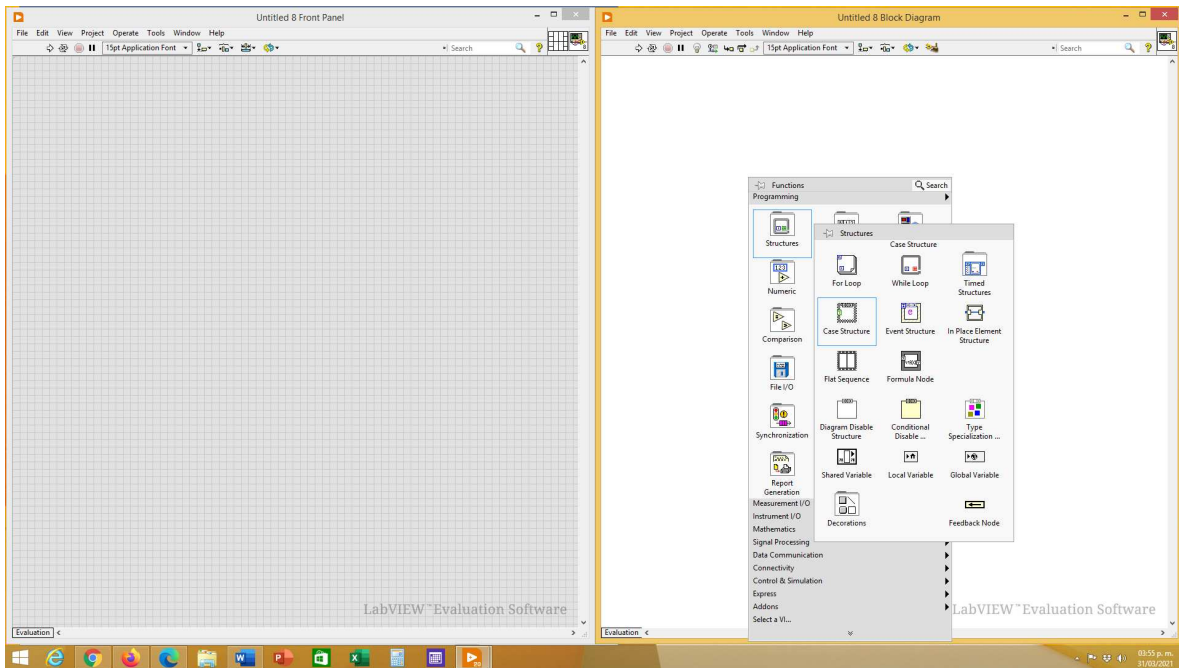
Si cambiamos los valores de los numeros A y B, veremos que en cada una de las salidas se muestra el resultado de la operación correspondiente



PRACTICA 10: APLICACIÓN USANDO ESTRUCTURA CASE

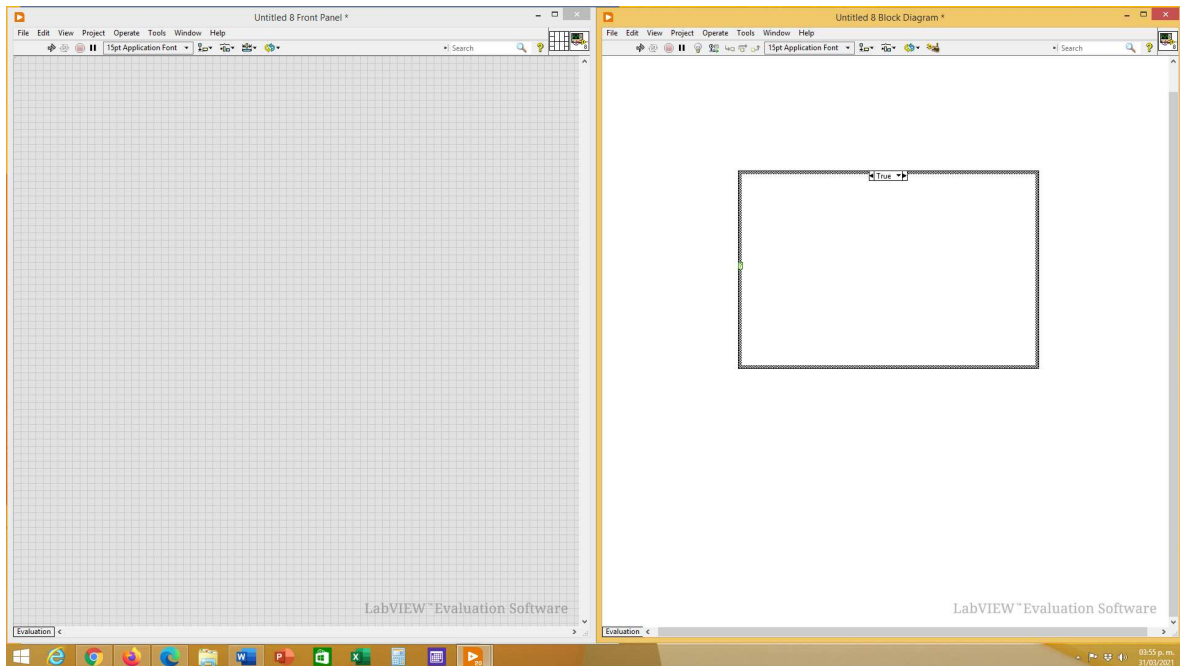
Abrimos un proyecto nuevo dando clic en Blank VI

Después clic con botón derecho en la ventana de Bloques para agregar en el Menu Structures, una estructura llamada Case Structure, esto se muestra en la imagen

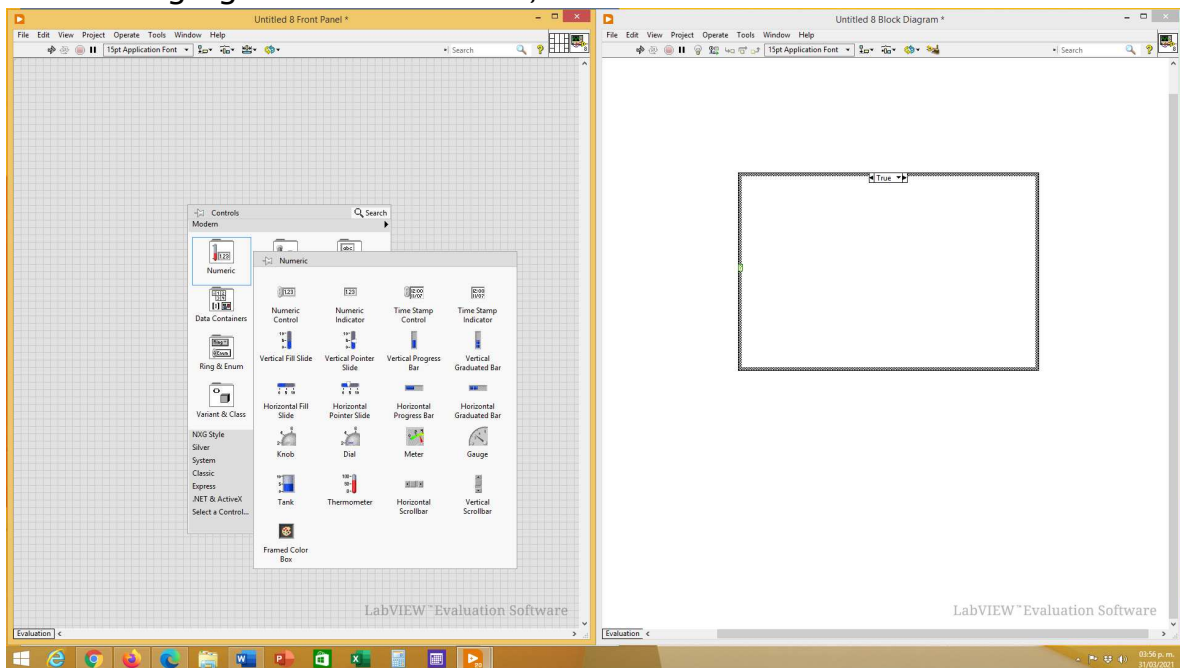


Aparecerá una ventanita con valor de TRUE. Si damos clic en las flechas, veremos la opción FALSE.

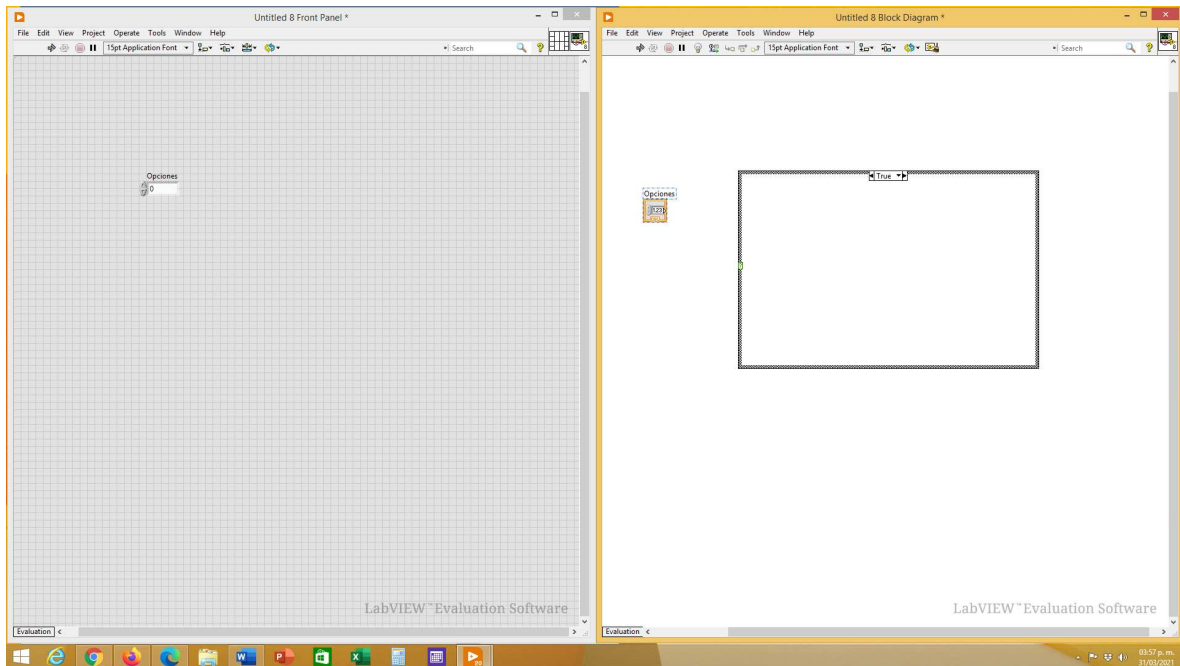
Esto lo usaremos en prácticas más adelante. Aquí usaremos opciones numéricas para poner 4 casos diferentes



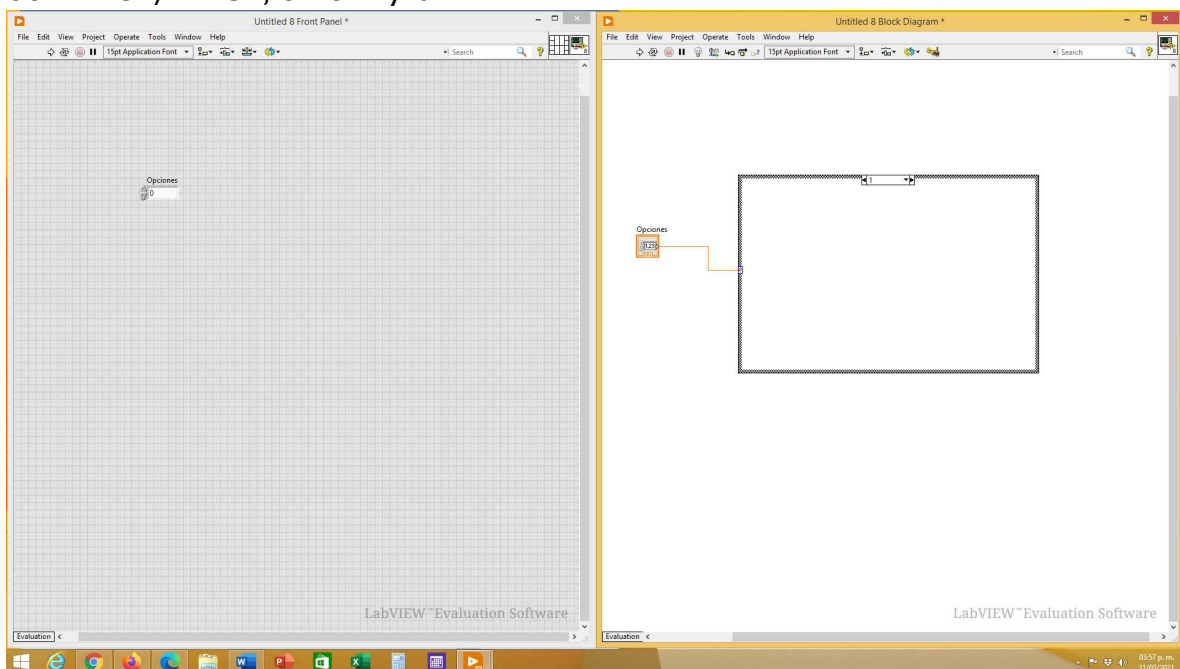
Vamos a agregar en Panel Frontal, un control numerico



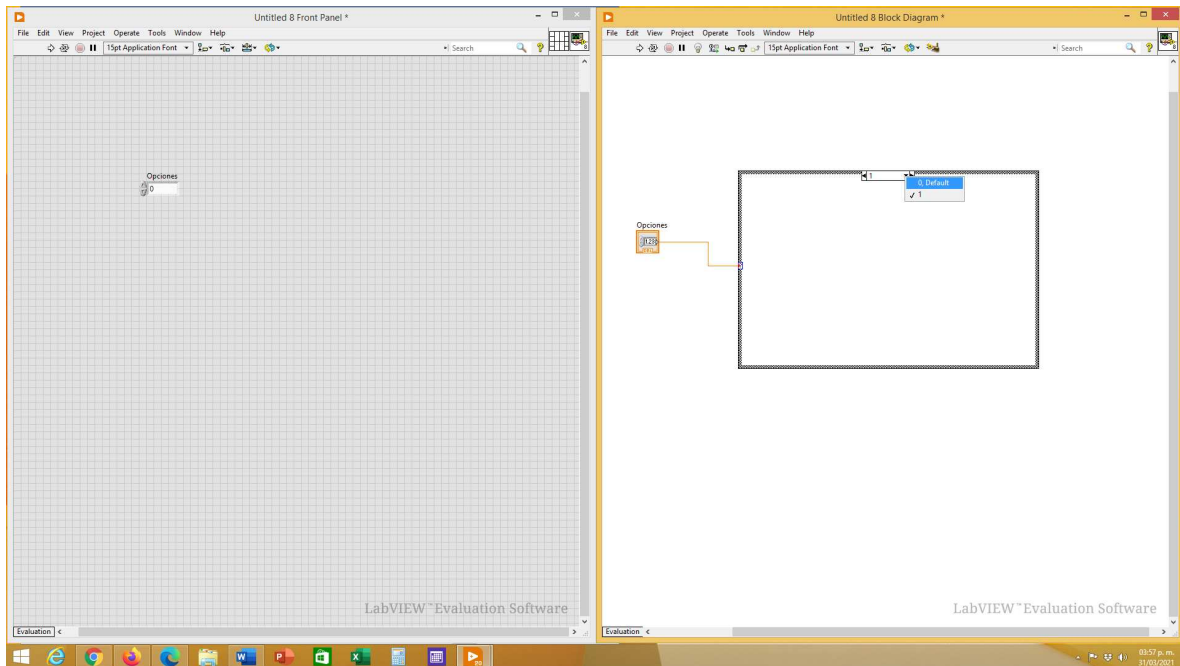
A este bloque le pondremos como nombre Opciones y quedaria asi



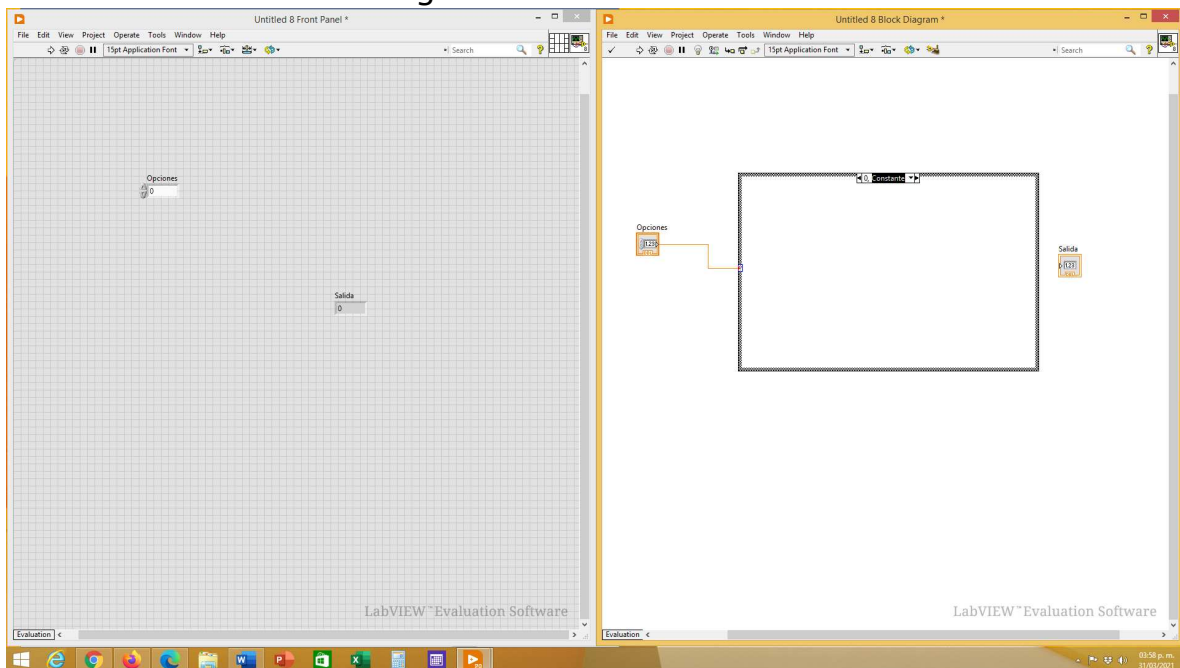
Conectaremos ese bloque de control al icono de interrogación (?) que aparece en el marco del CASE. Si observamos, las opciones del case ya no son TRUE/FALSE, sino 1 y 0



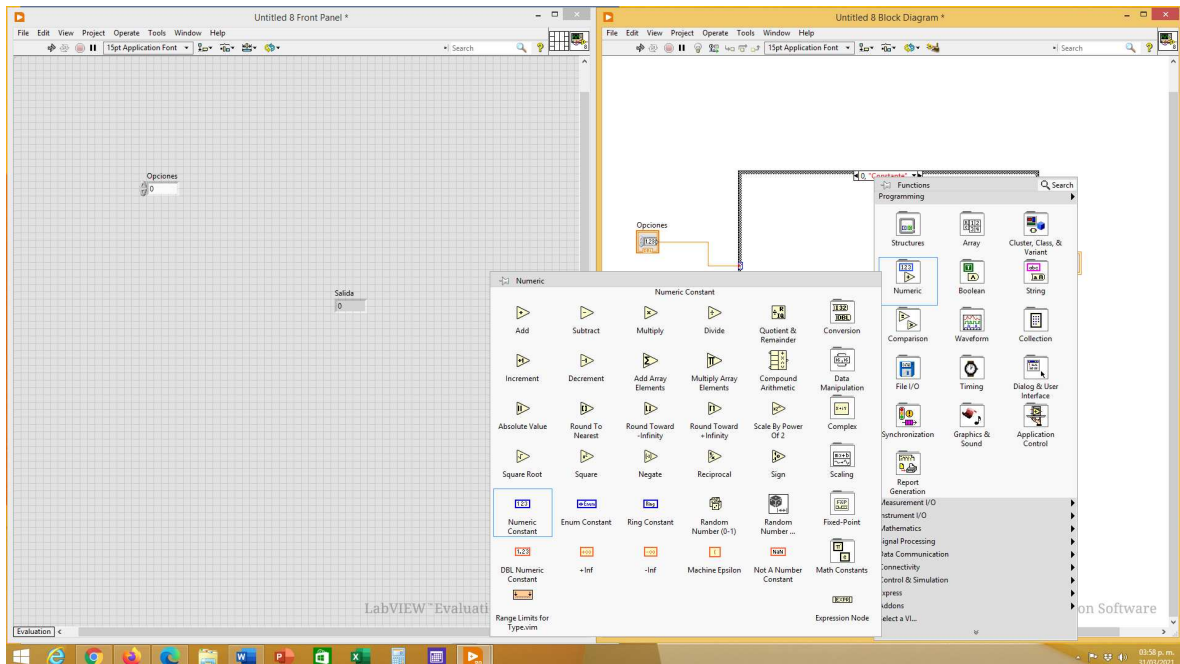
Por default, el valor es de 0 como se observa



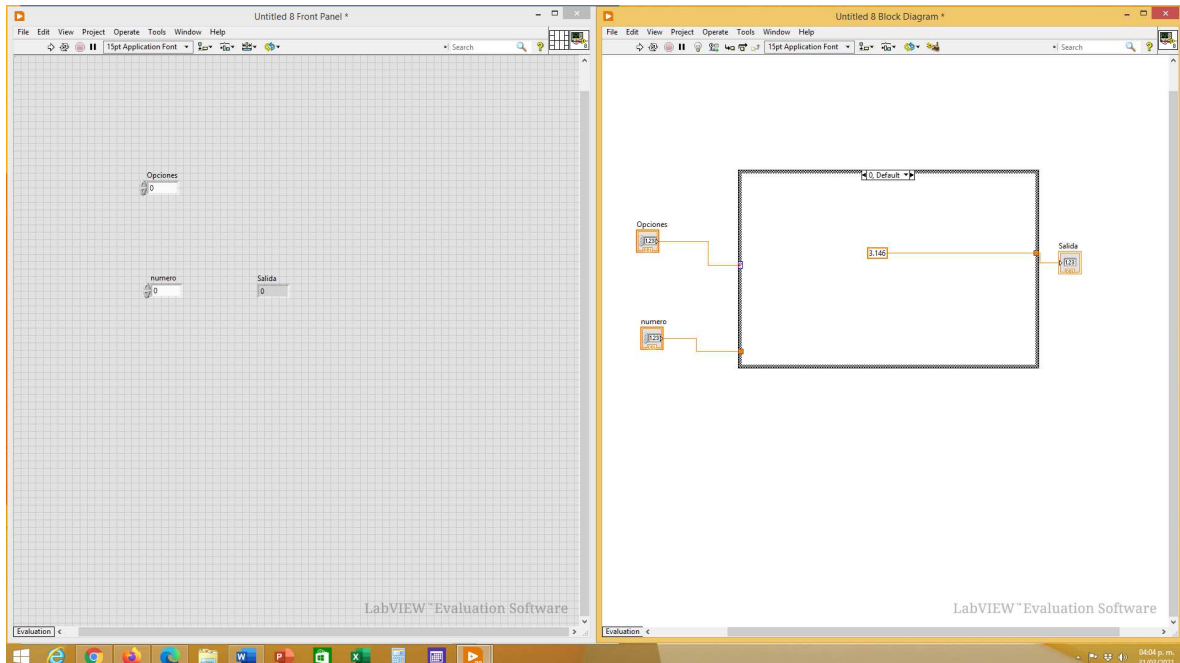
Dando clic en la opcion 0, vamos a colocar un visualizador numerico y lo colocaremos como en la figura



Metermos una constante numerica y le damos cualquier valor

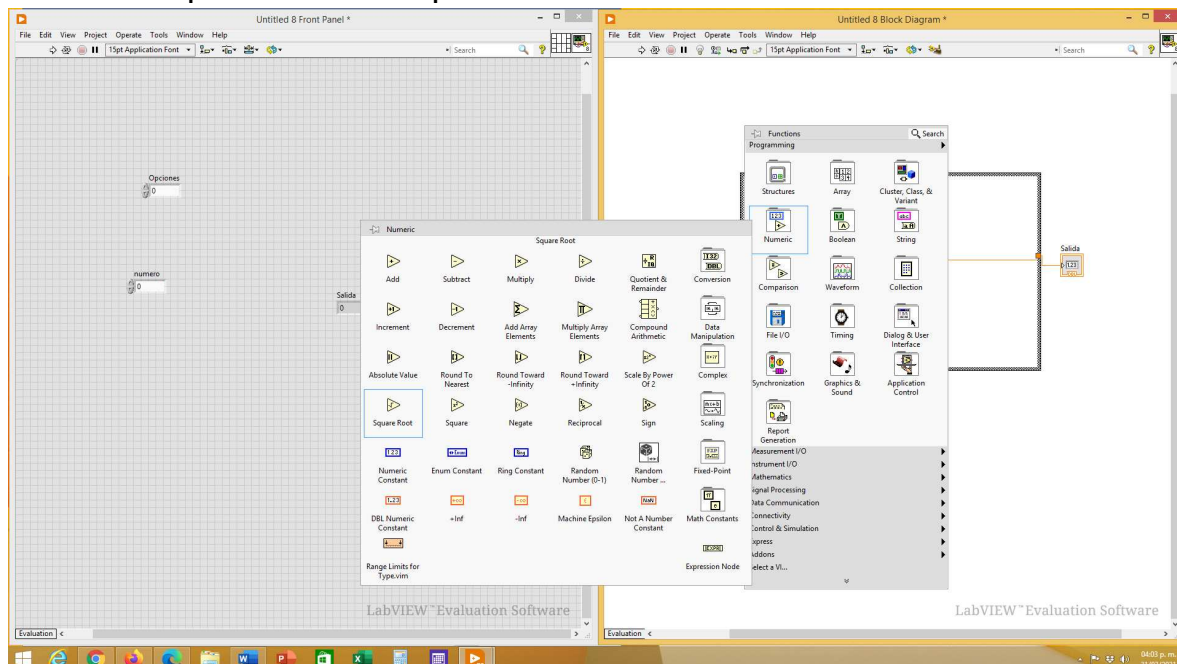


Esta constante debera estar dentro del cuadro del CASE, verificando que la opcion sea 0, Default y conectamos el bloque de la constante directamente al bloque visor de salida.

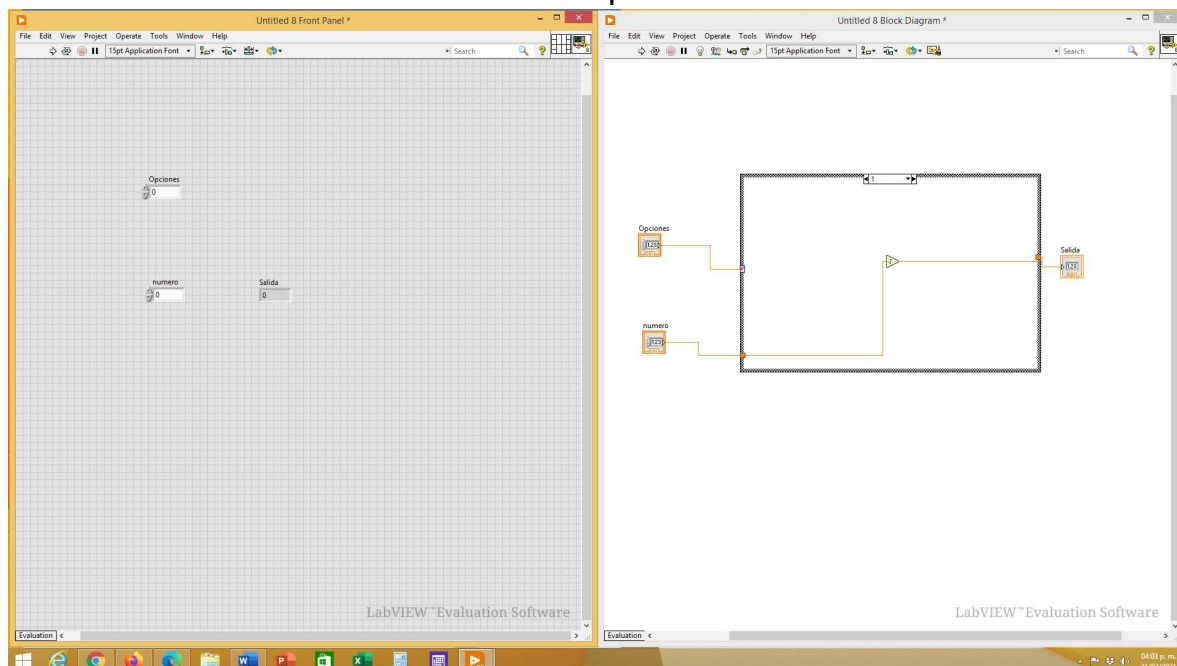


Si ahora cambiamos ahora la opcion por el valor 1 en el case, haremos que ahora se muestre la raiz cuadrada de un numero ingresado desde un control numerico

Para esto metemos otro control numerico que llamaremos numero, junto con un bloque Numerico Square Root

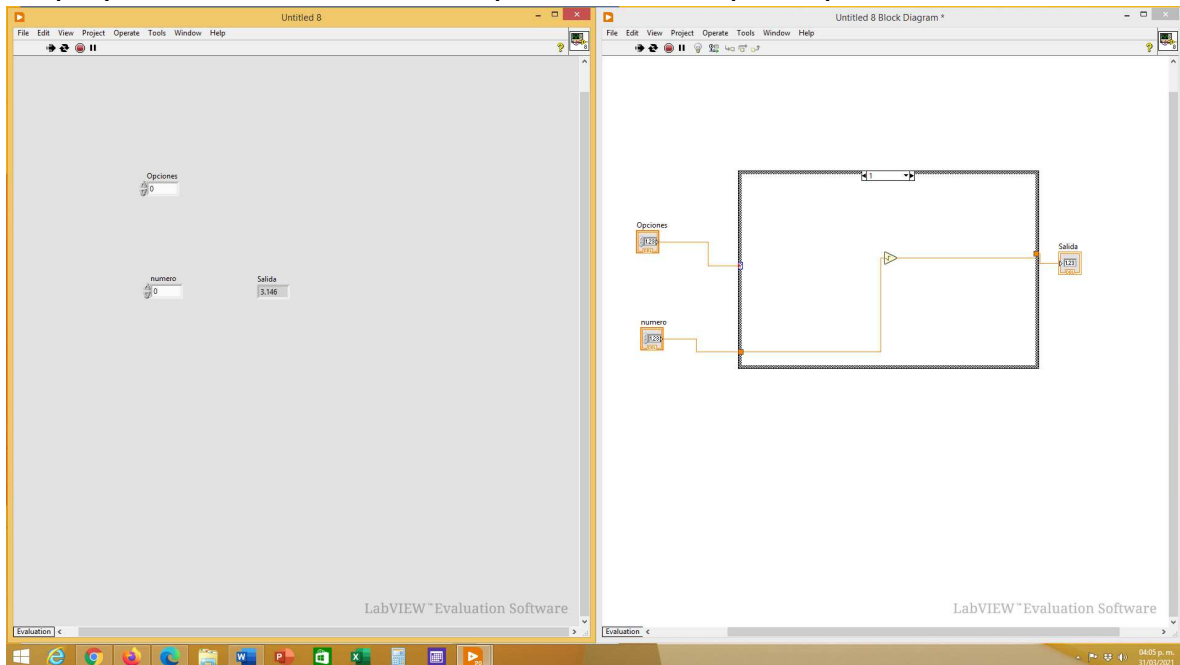


Seleccionado en el Case, el valor de 1, conectamos el control llamado numero y lo metemos al Square Root y la salida de este bloque lo conectamos al visualizador numerico que se encuentra a la derecha

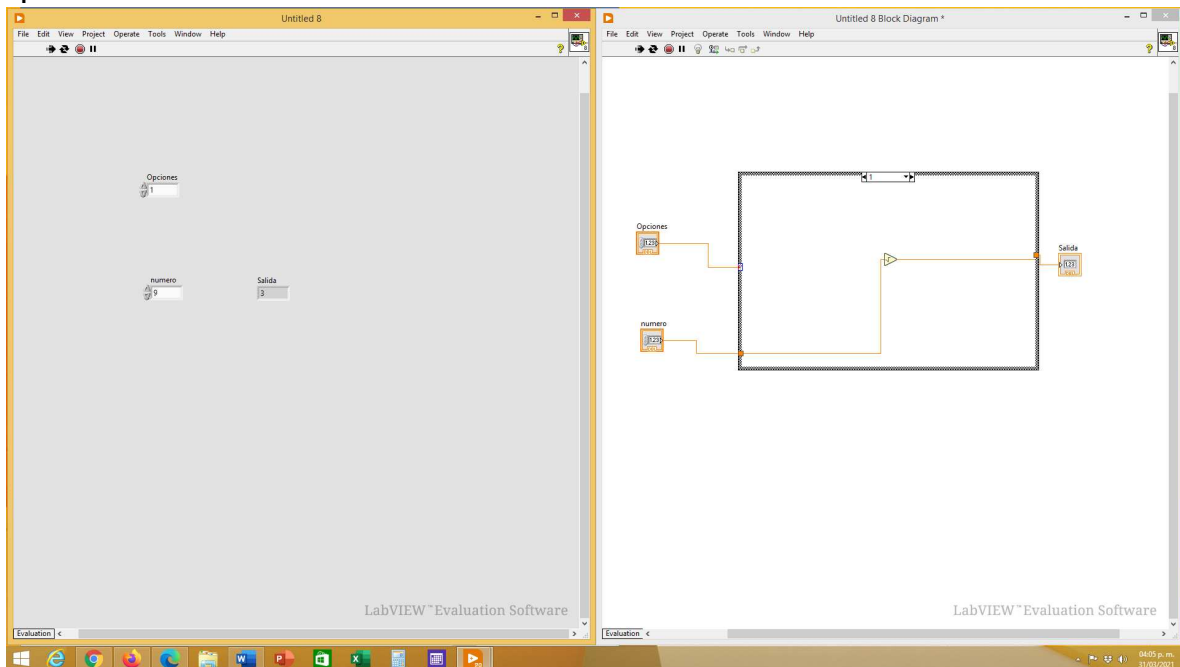


Ejecutamos dando clic en las flechas blancas

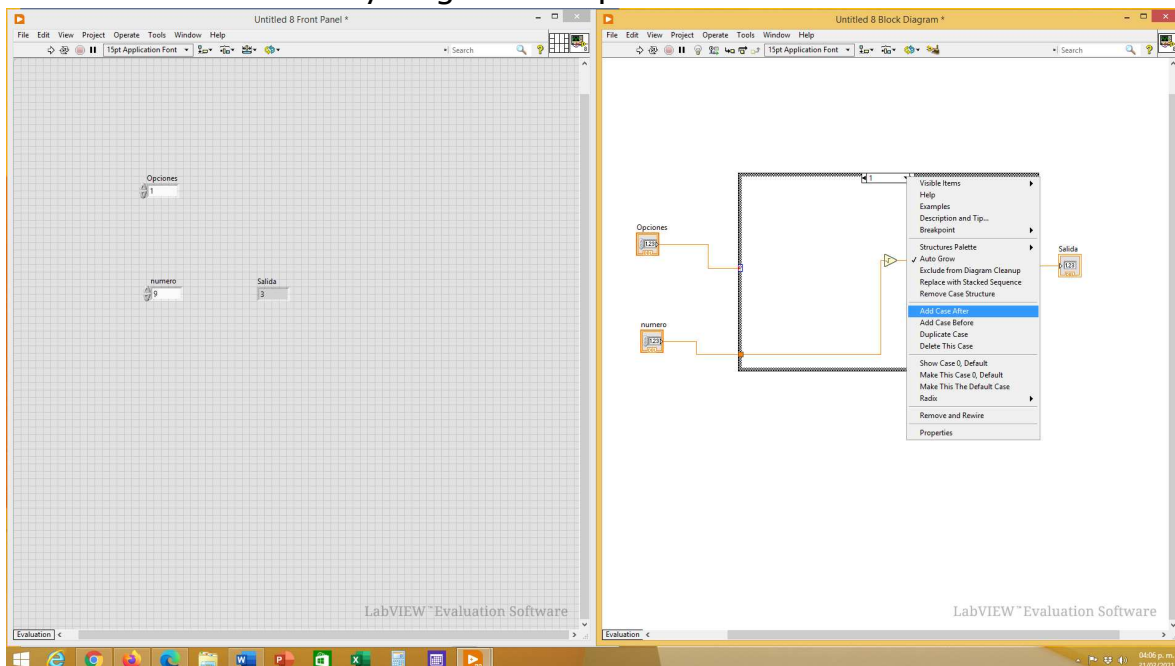
Desde el control numerico elegimos 0 y observemos que aparece en el display Salida el valor de 3.14 que es el valor que le pusimos a la variable



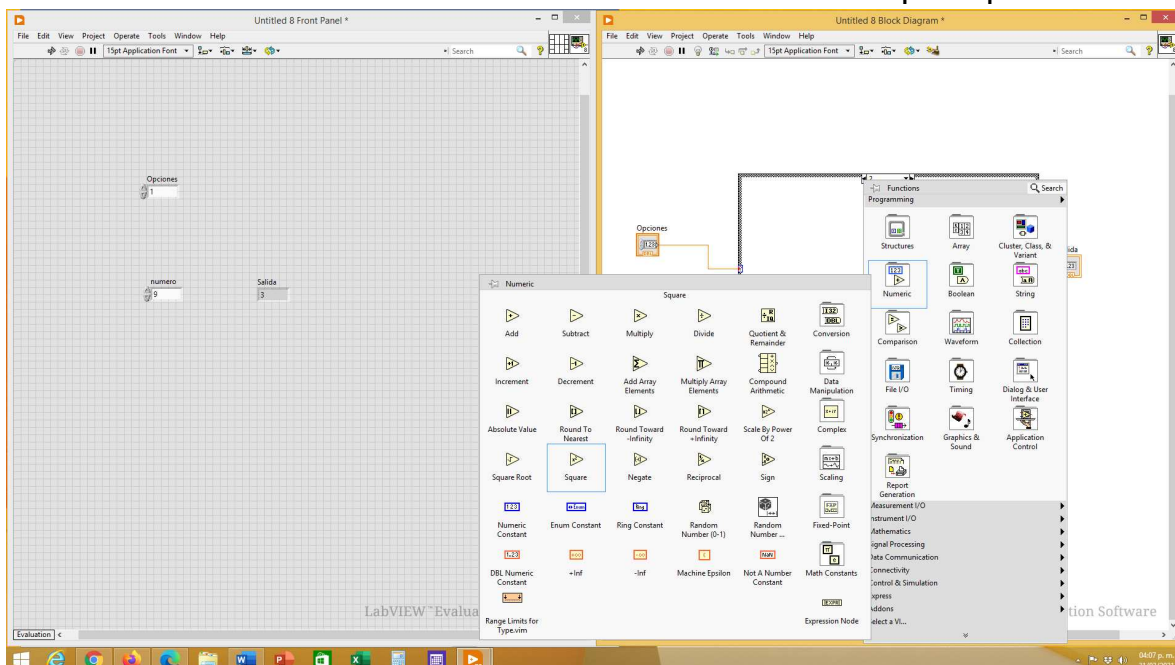
Desde el control numerico elegimos 1 y observemos que aparece en el display Salida el valor de 3 que seria la raiz cuadrada del numero ingresado que es el 9



Hasta ahora, hemos realizado 2 opciones y en el CASE no hay mas que 0 y 1, pero si le queremos agregar algunas mas, sobre el control del CASE damos clic secundario y elegimos la opcion Add Case after

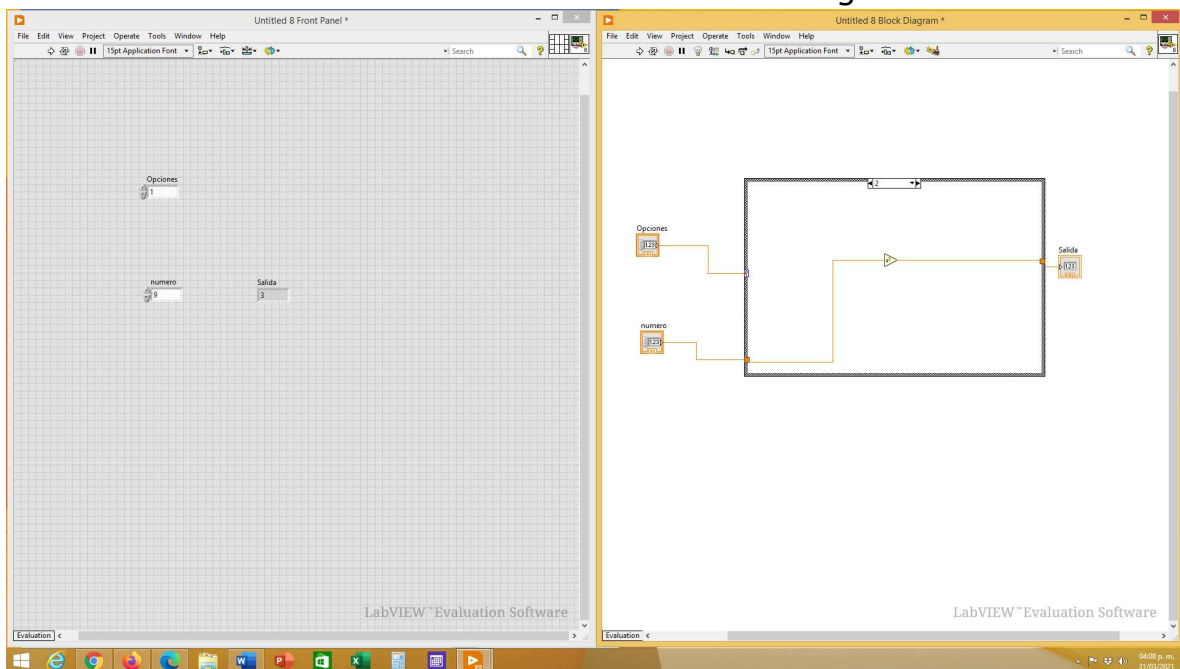


Aparecera en el CASE ahora tambien la opcion 0, 1 y 2
En esta opcion 2 meteremos un numero para que nos muestre en la salida el cuadrado de este mismo. Para esto necesitamos el bloque Square

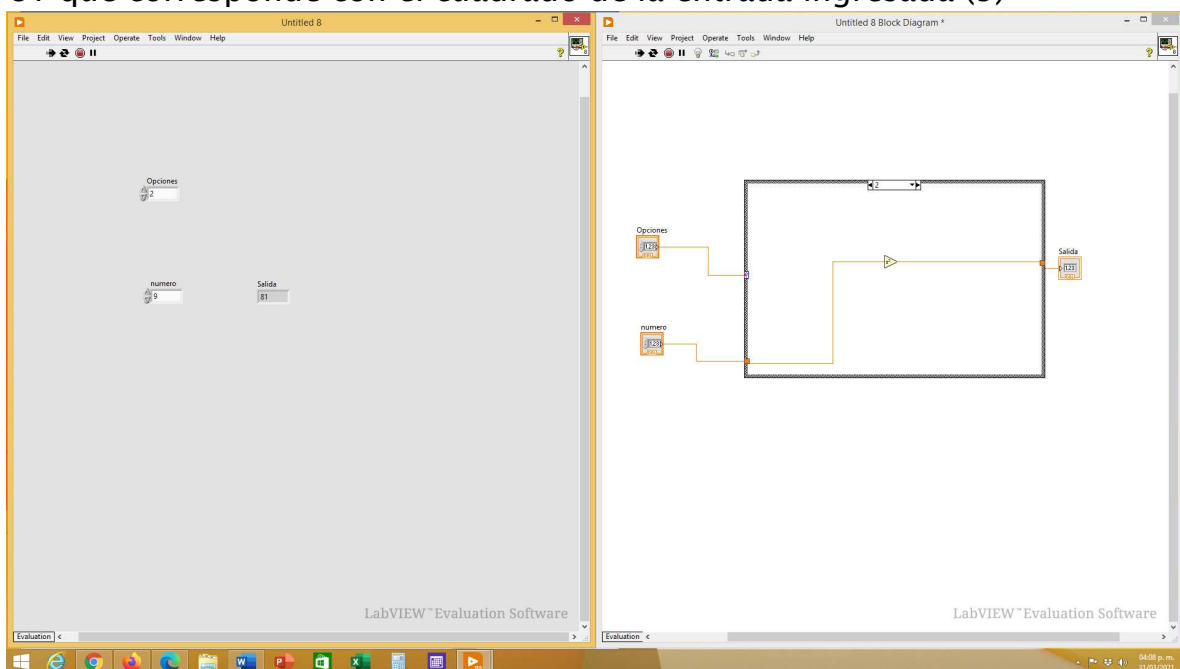


Observemos que ahora debemos estar trabajando en el CASE 2.

Conectamos los controles como se muestra en la imagen



Y al ejecutar, aparecera la 3ra opcion (numero 2) y en la salida veremos un 81 que corresponde con el cuadrado de la entrada ingresada (9)

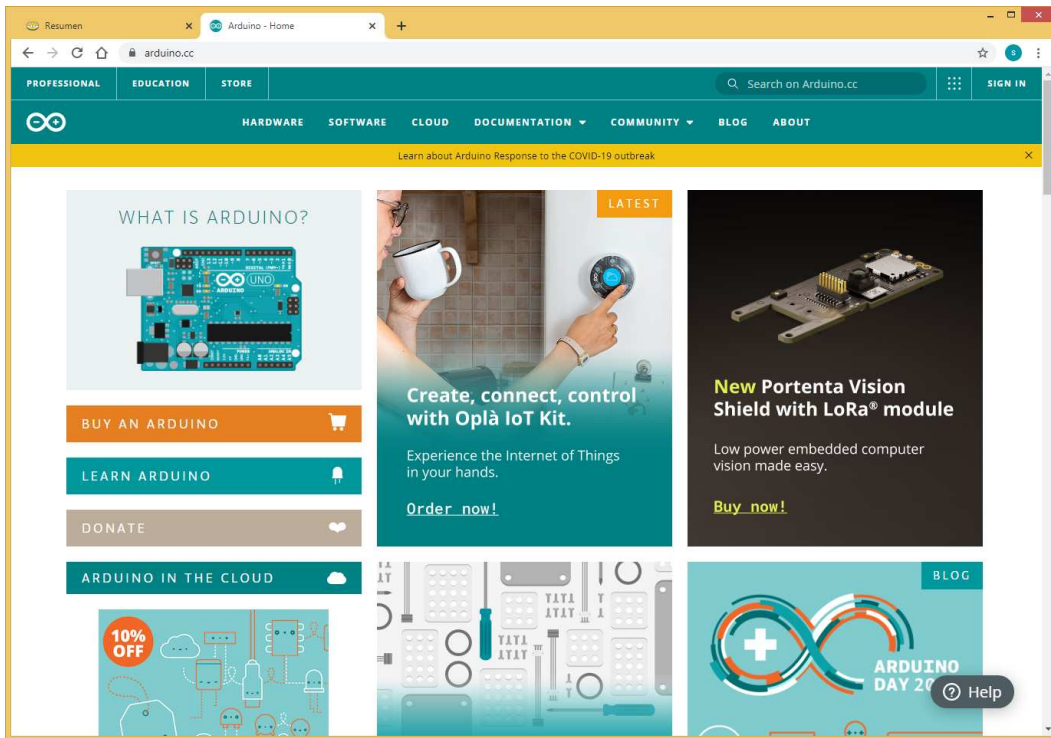


Practica 11 Arduino con LabView

En este apartado veremos la comunicación serial entre LabView y Arduino para controlar una salida digital

Para ello, necesitaremos tener instalado Arduino. Si no contamos con Arduino, ingresamos a la pagina arduino.cc y en el menu SOFTWARE damos

Clic



En Downloads, buscaremos la versión de Windows, Linux o macOS X que necesitamos. En caso de Windows necesitamos dar clic en la primera opción que se muestra en la pantalla.

Resumen x Software | Arduino x +

arduino.cc/en/software

PROFESSIONAL EDUCATION STORE

SEARCH on Arduino.cc

SIGN IN

HARDWARE SOFTWARE CLOUD DOCUMENTATION COMMUNITY BLOG ABOUT

Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

CODE ONLINE GETTING STARTED

10% OFF

Arduino Day Special Deals

Downloads

 **Arduino IDE 1.8.13**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 [Get](#)

Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Mac OS X 10.10 or newer

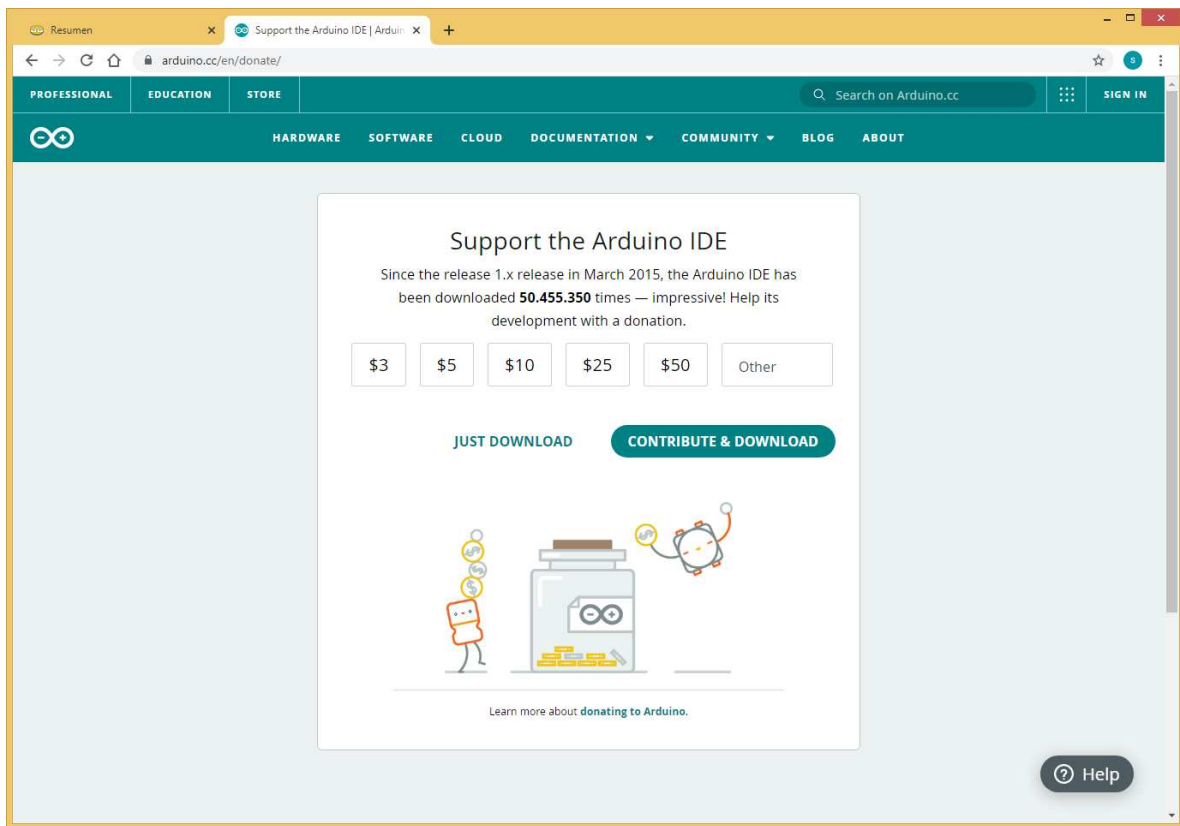
Release Notes Checksums (sha512)

[Help](#)

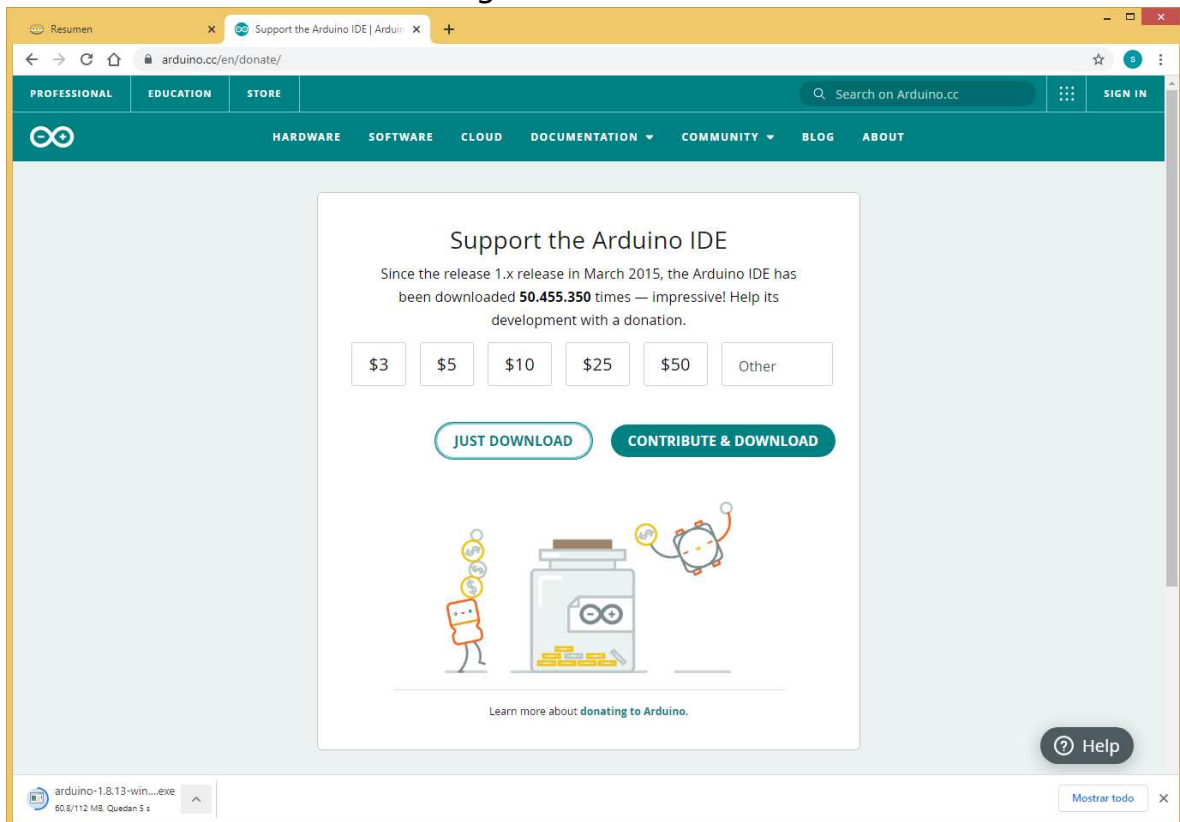
<https://www.arduino.cc/en/software>

Hourly Builds Previous Releases

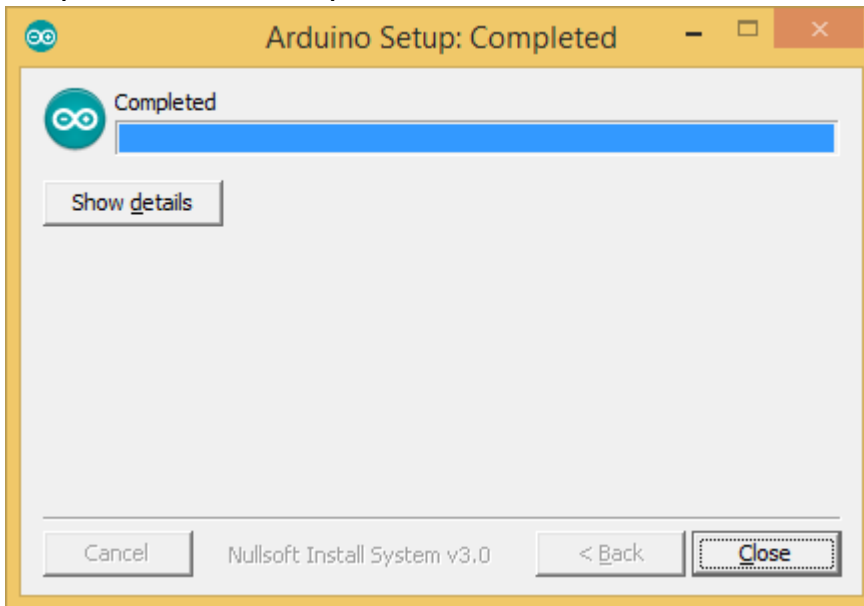
La siguiente ventana es para donaciones, únicamente daremos clic en Just download.



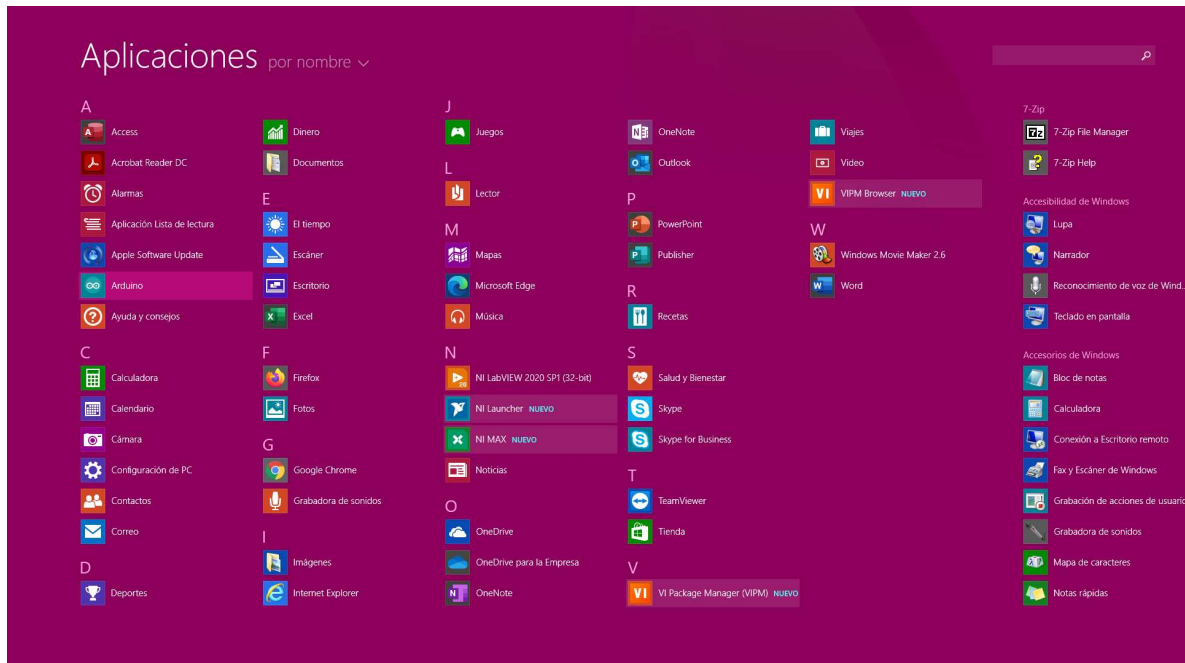
El archivo comenzará a descargarse



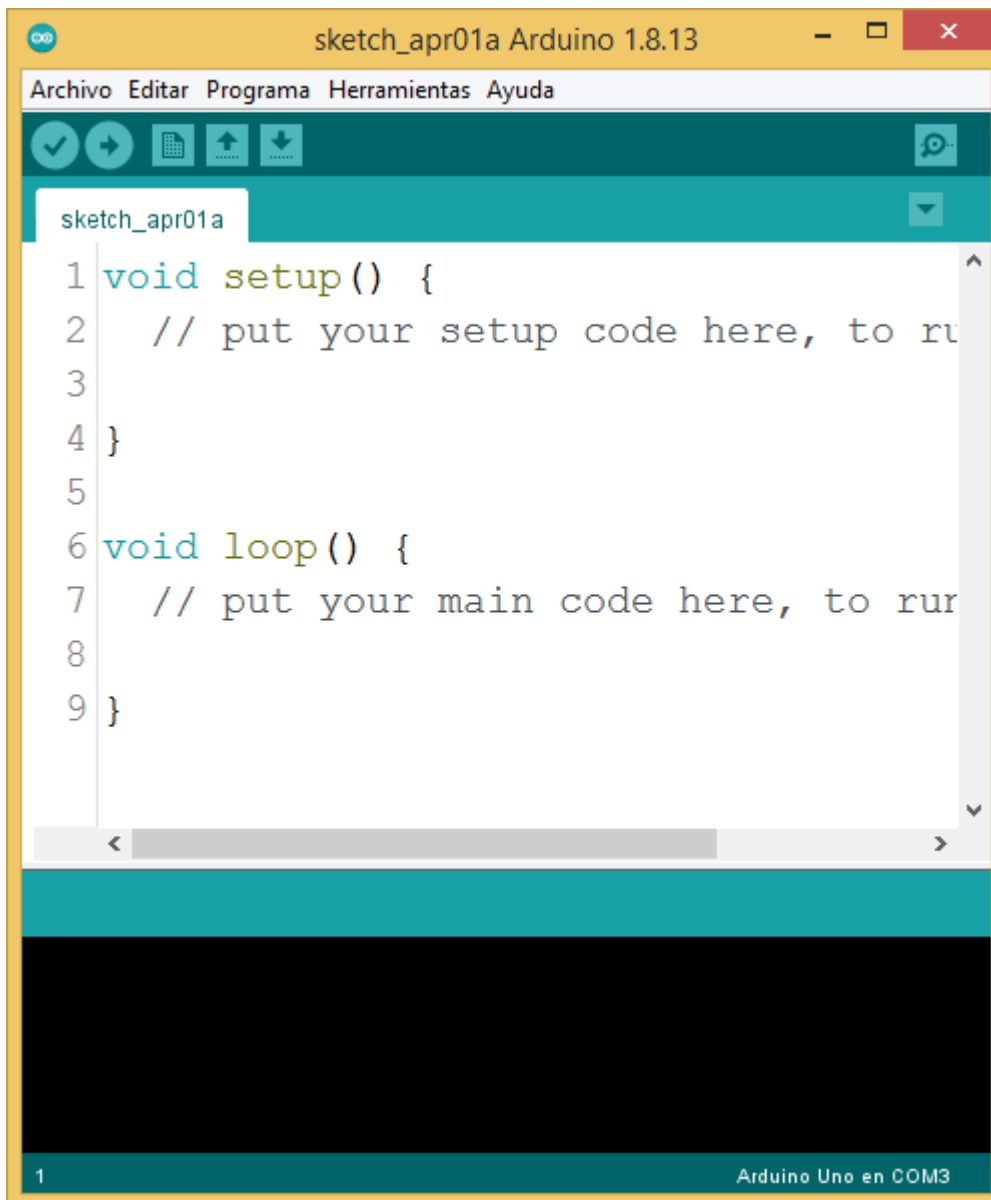
Una vez descargado si da doble clic para instalarlo y en la mayoría de las ventanas del asistente damos clic en siguiente hasta finalizar la instalación aceptando todas las pantallas.



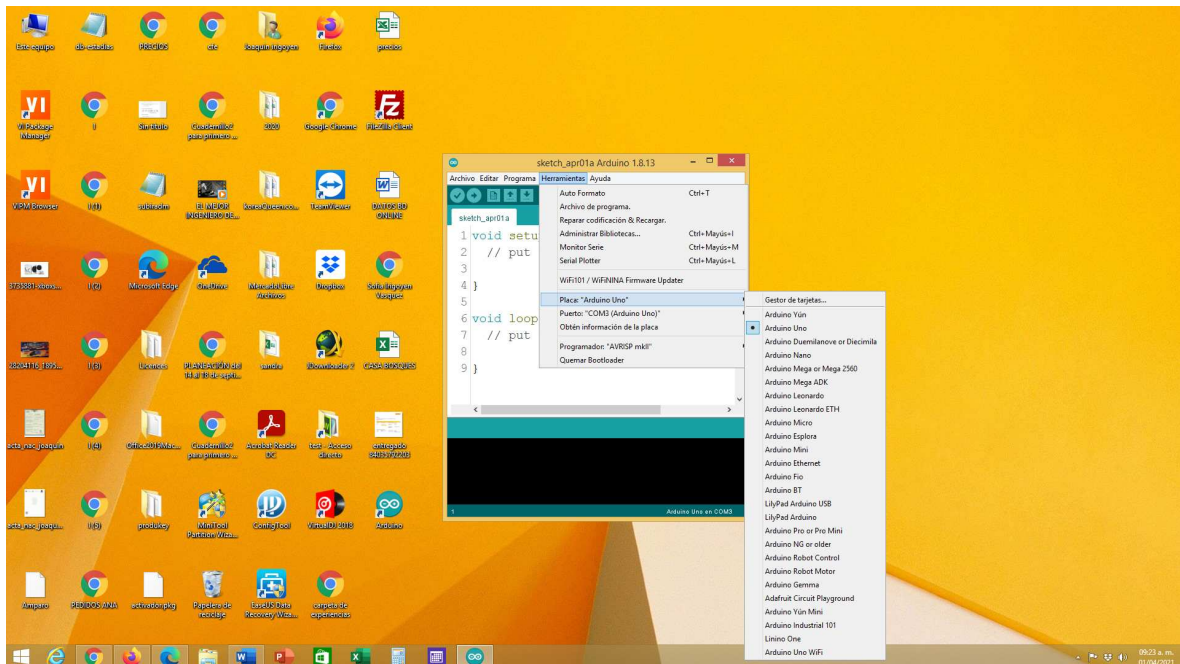
Nuestra aplicación de Arduino la encontraremos en los programas instalados



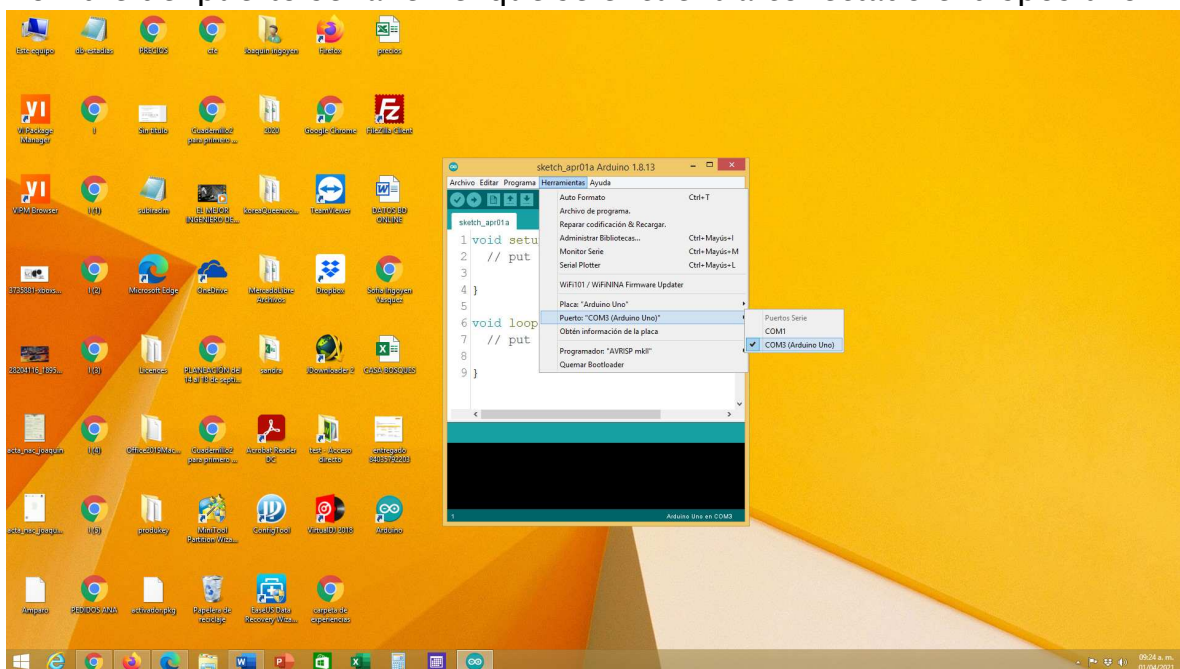
Al abrir la aplicación se mostrará un código de ejemplo



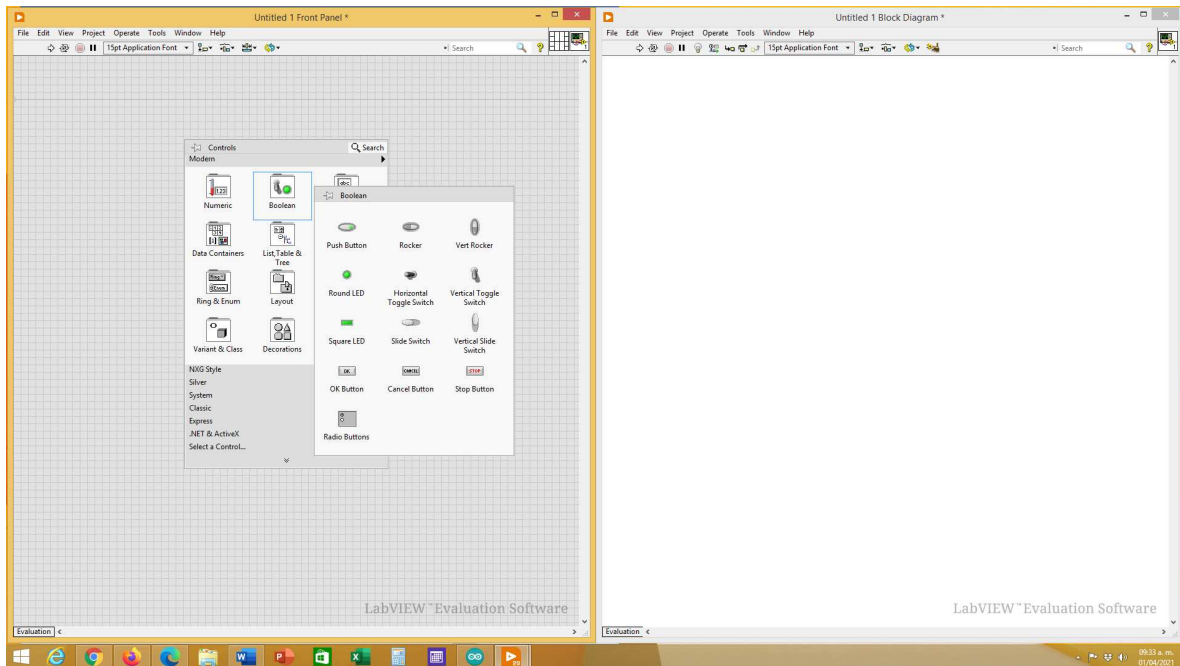
Con el Arduino conectado, damos clic en el menú herramientas y en la opción de placa debemos seleccionar la versión que tengamos conectada, en este caso será Arduino uno



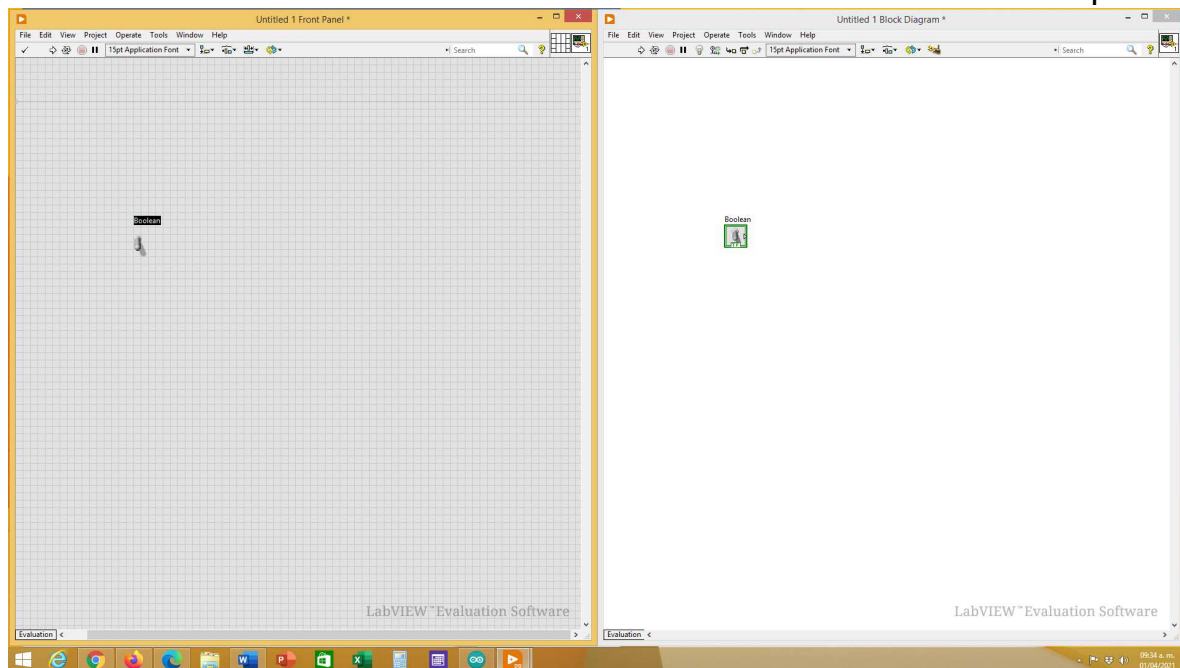
Posteriormente en el mismo menú, seleccionamos el puerto y damos clic donde aparezca entre paréntesis Arduino UNO. Debemos fijarnos en el nombre del puerto serial en el que se encuentra conectado el dispositivo



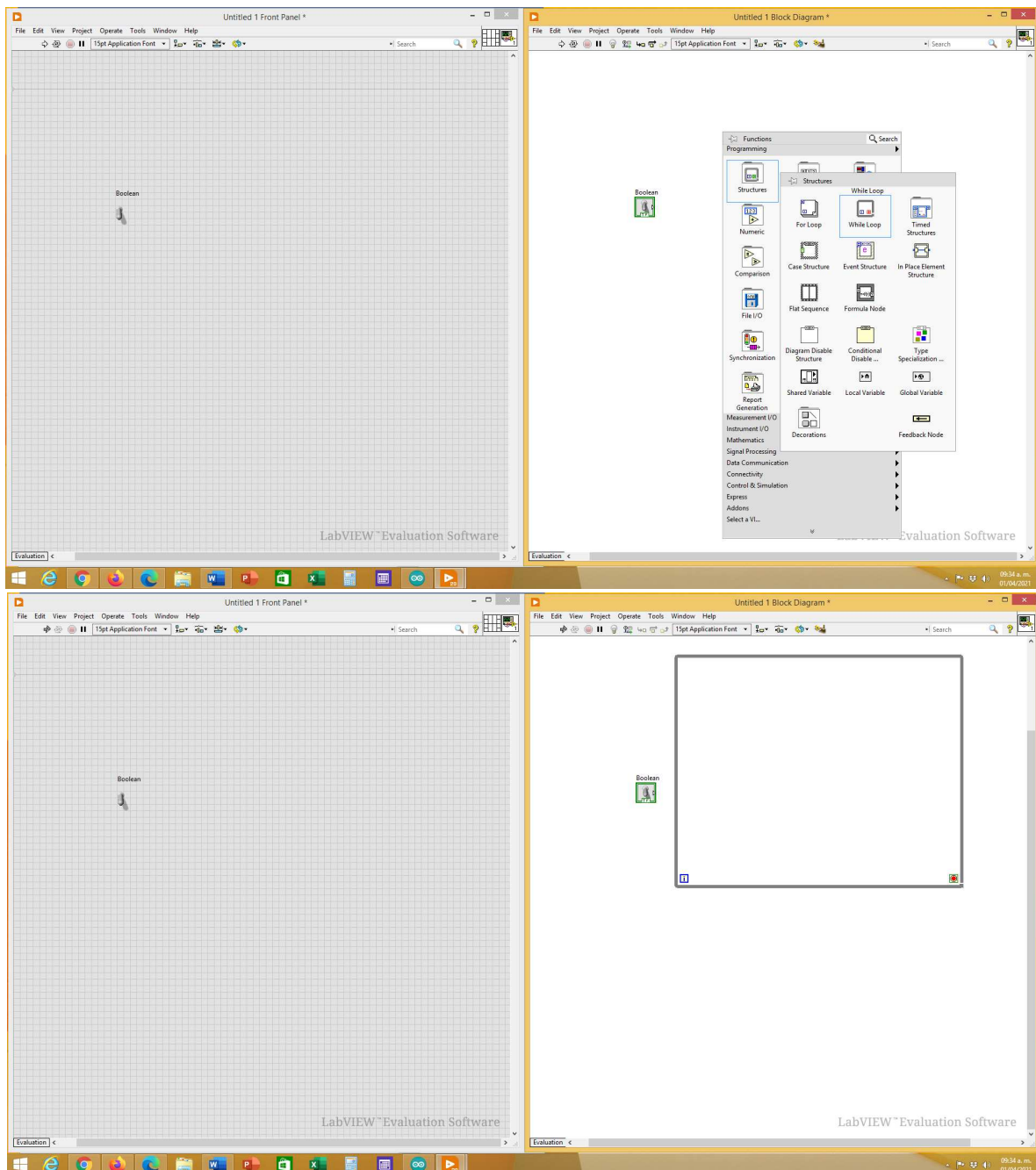
Regresando a LabVIEW, vamos a elegir en el panel frontal los controles boolean, Y elegimos vertical toggle switch



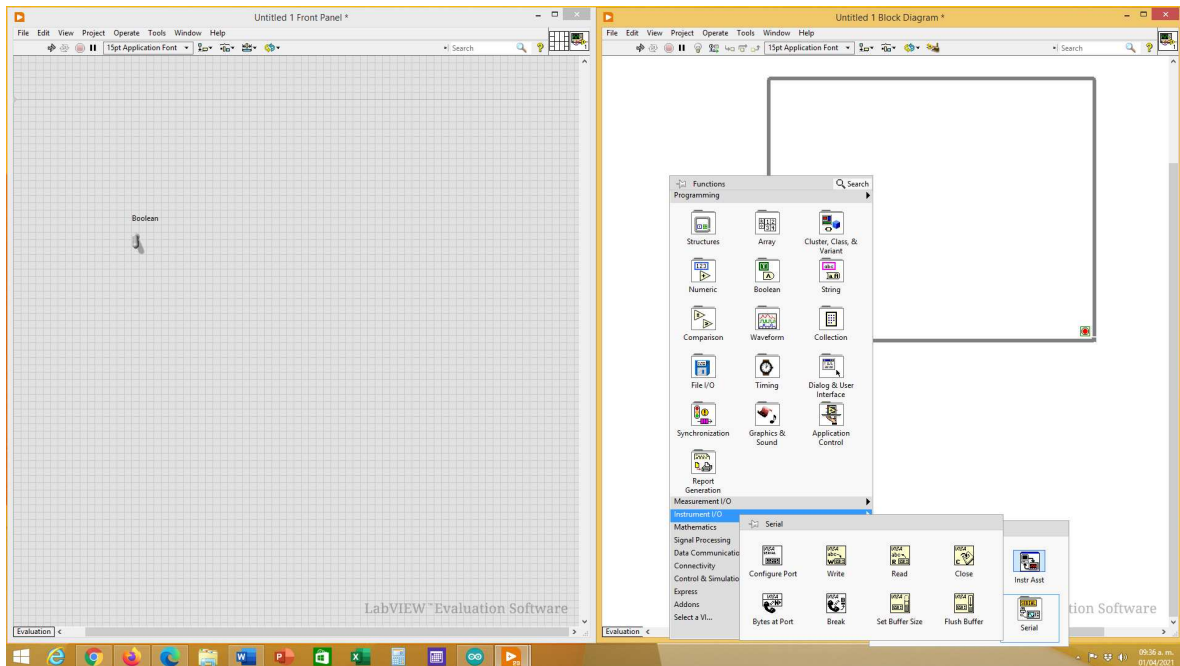
Y como se observa se debera mostrar también en la ventana de diagrama de bloques



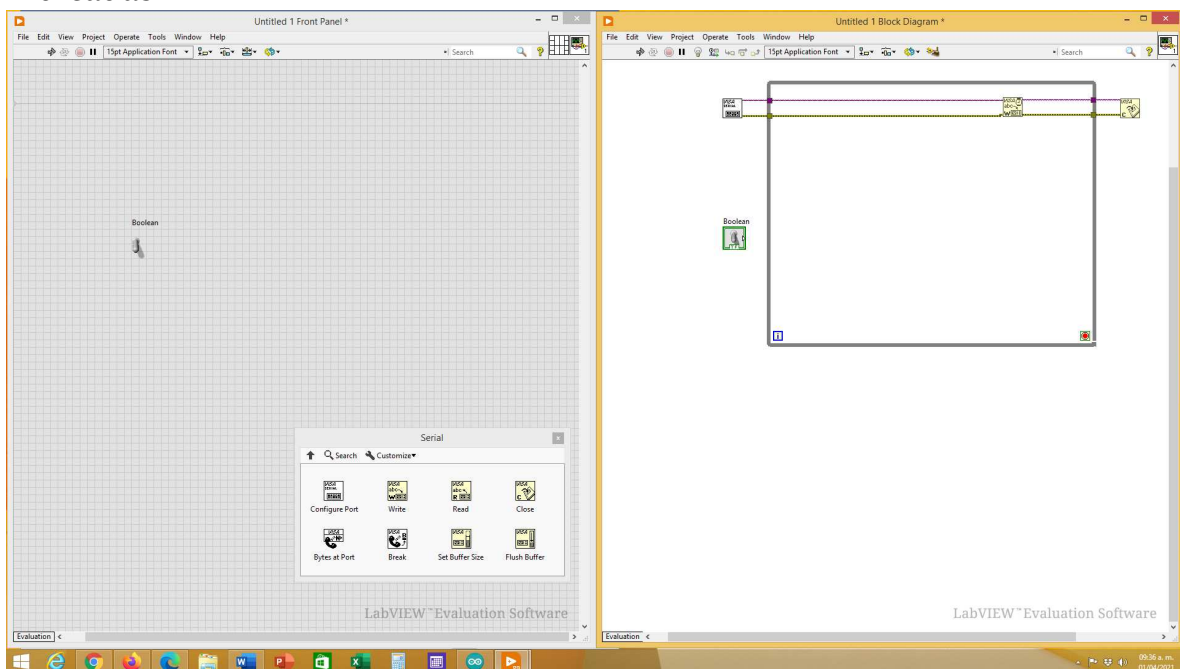
en esta ventana vamos a insertar una estructura tipo While loop



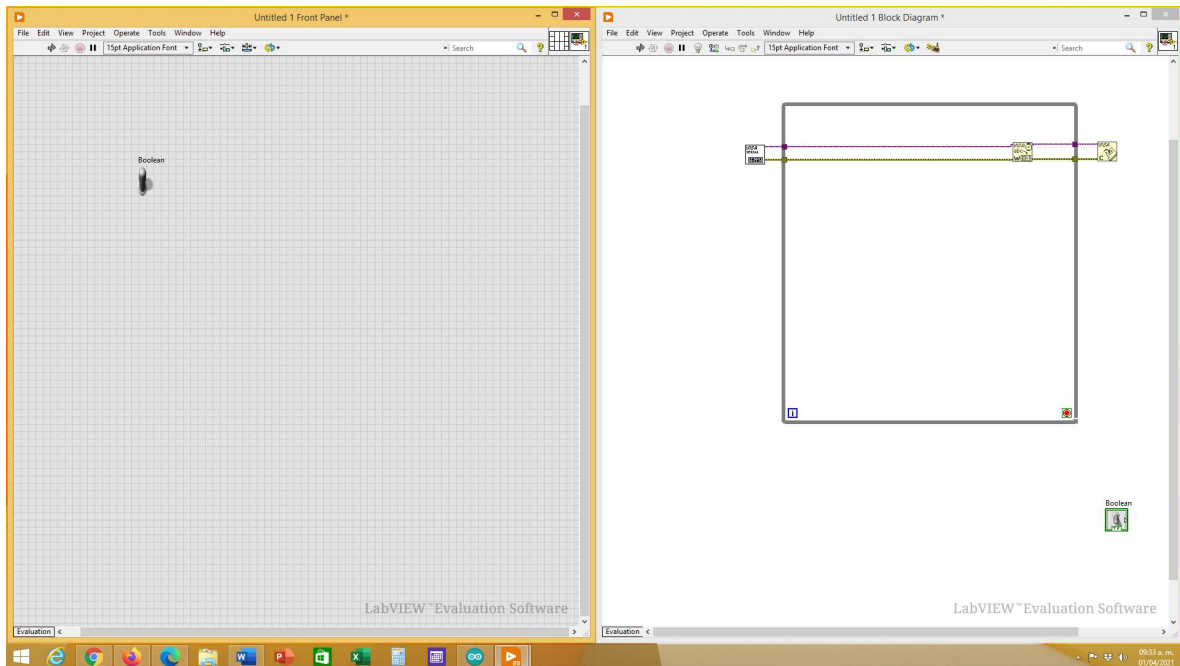
para realizar la comunicación con el puerto serial, vamos a el menú instrument I/O, damos clic en serial y clic en la tachuela para anclar nuestra ventana serial



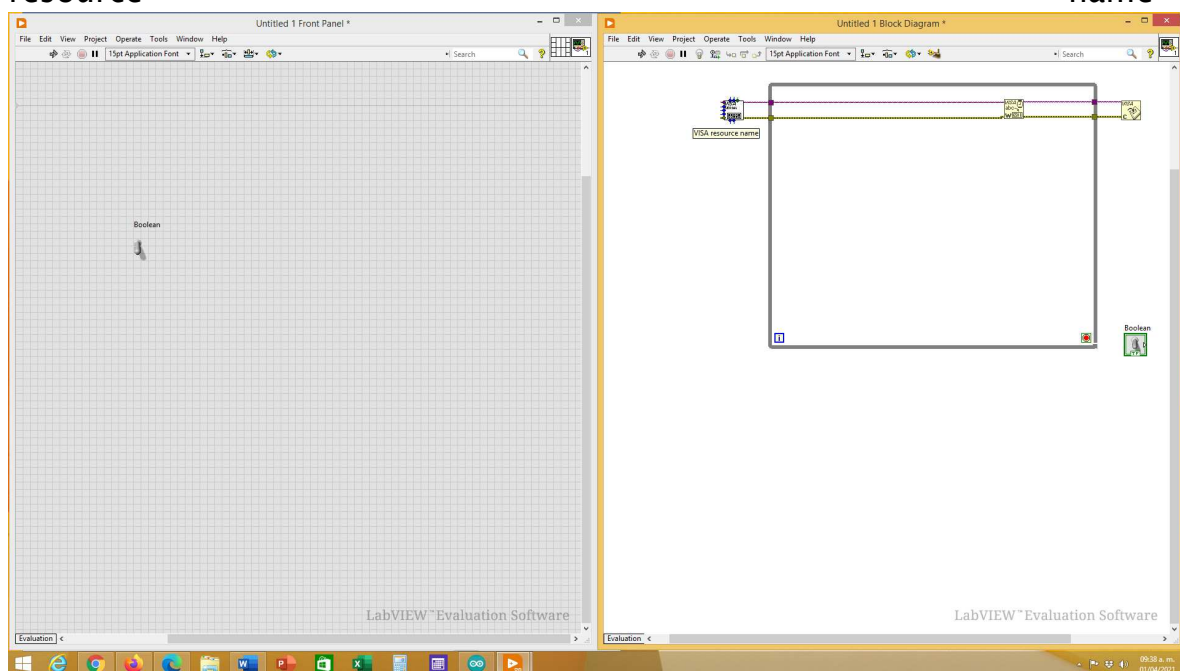
Aquí insertaremos los controles Configure PORT, Write y Close. Se deberán acomodar como se muestra en la imagen, siguiendo las conexiones indicadas.



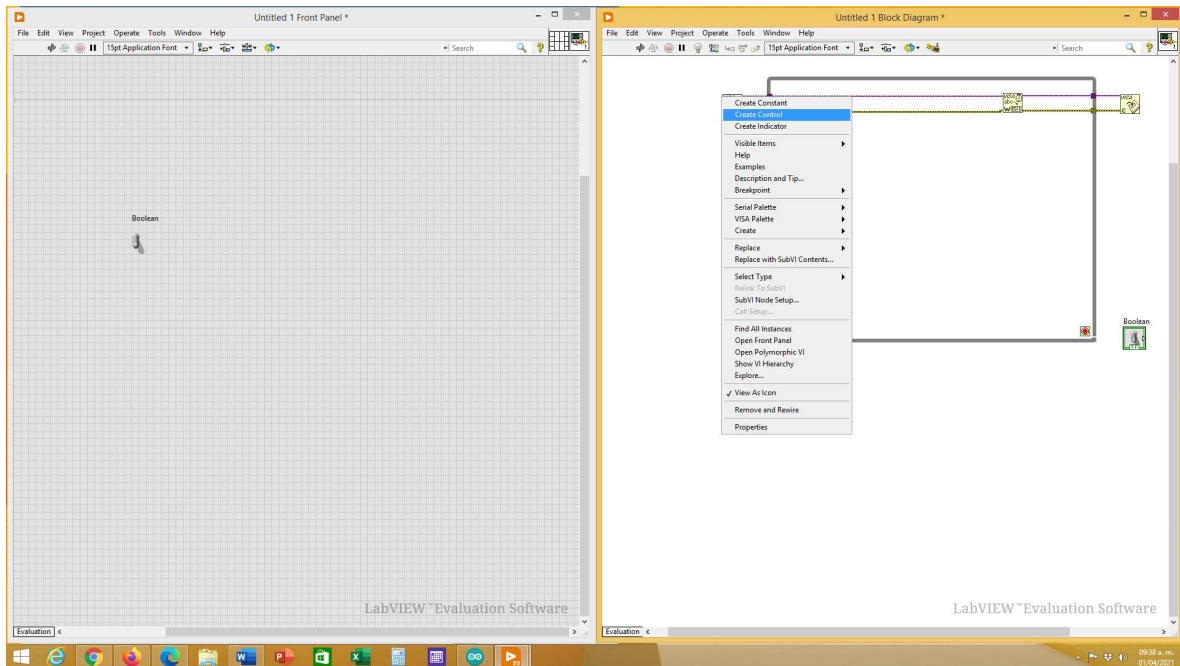
Aquí ya podemos cerrar la ventana de los controles Serial



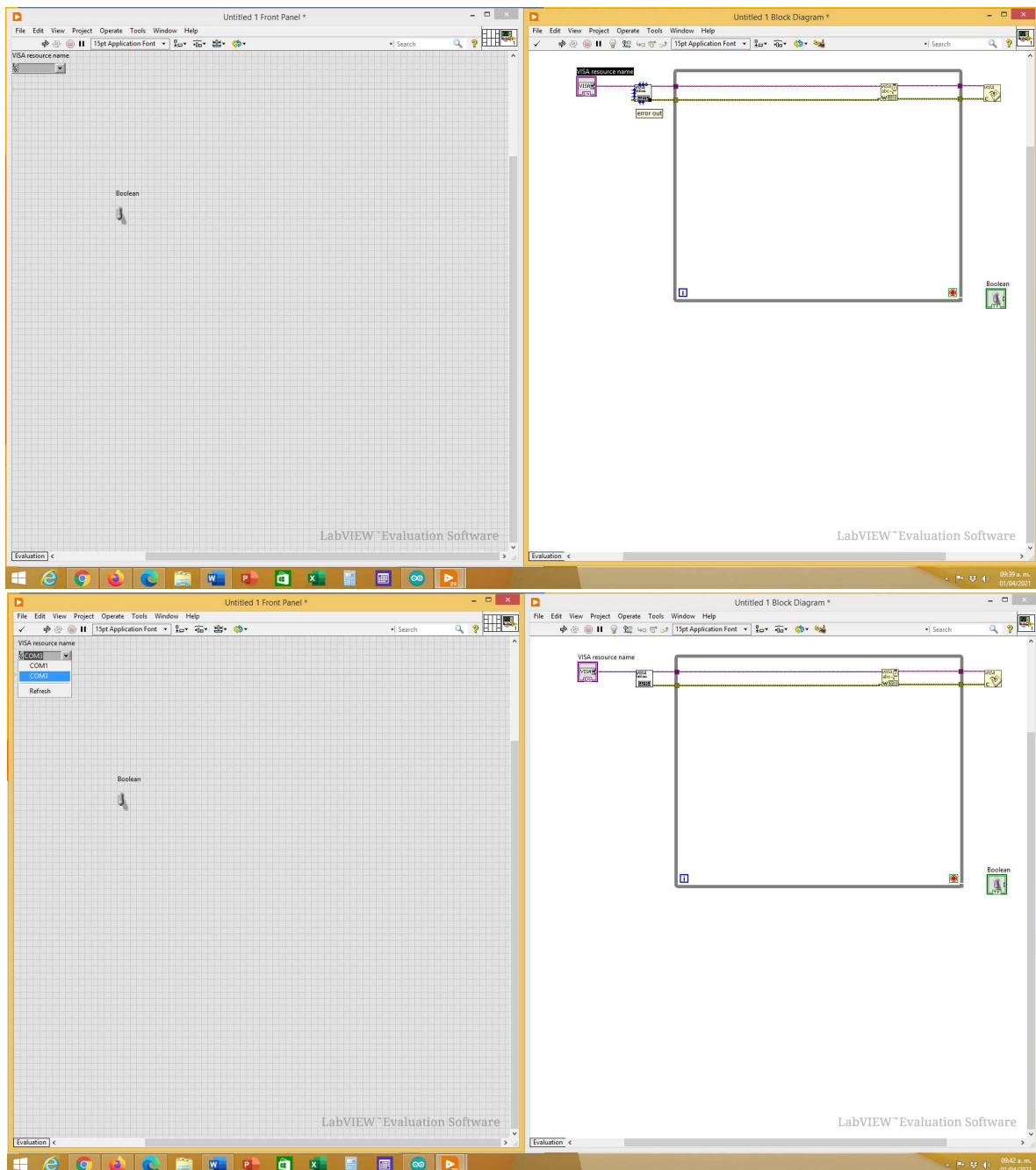
Para agregar un puerto de comunicación, daremos clic en el primer punto de arriba del control Configure Port, donde aparecera el texto VISA resource name



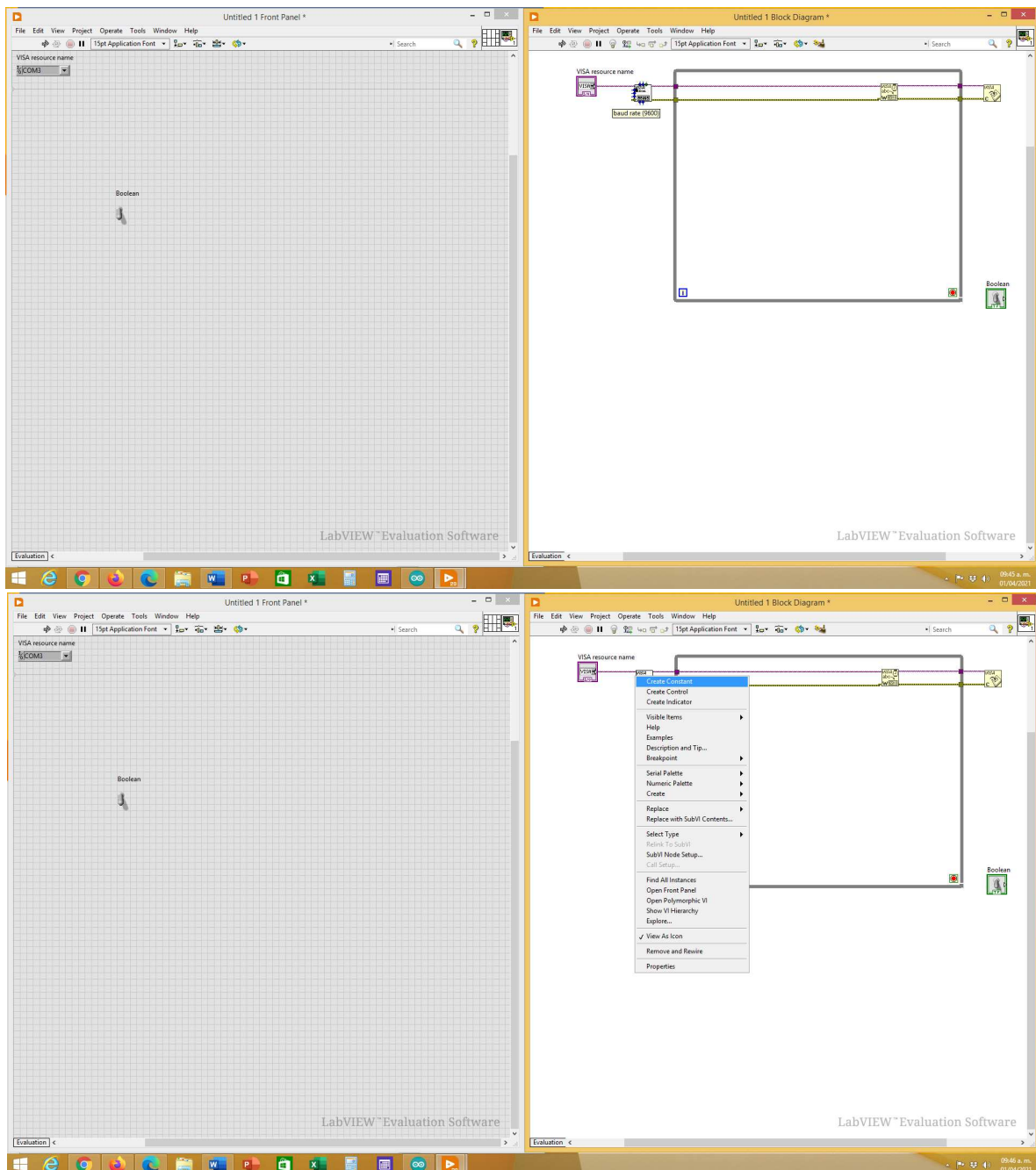
y damos clic en la opcion Create Control



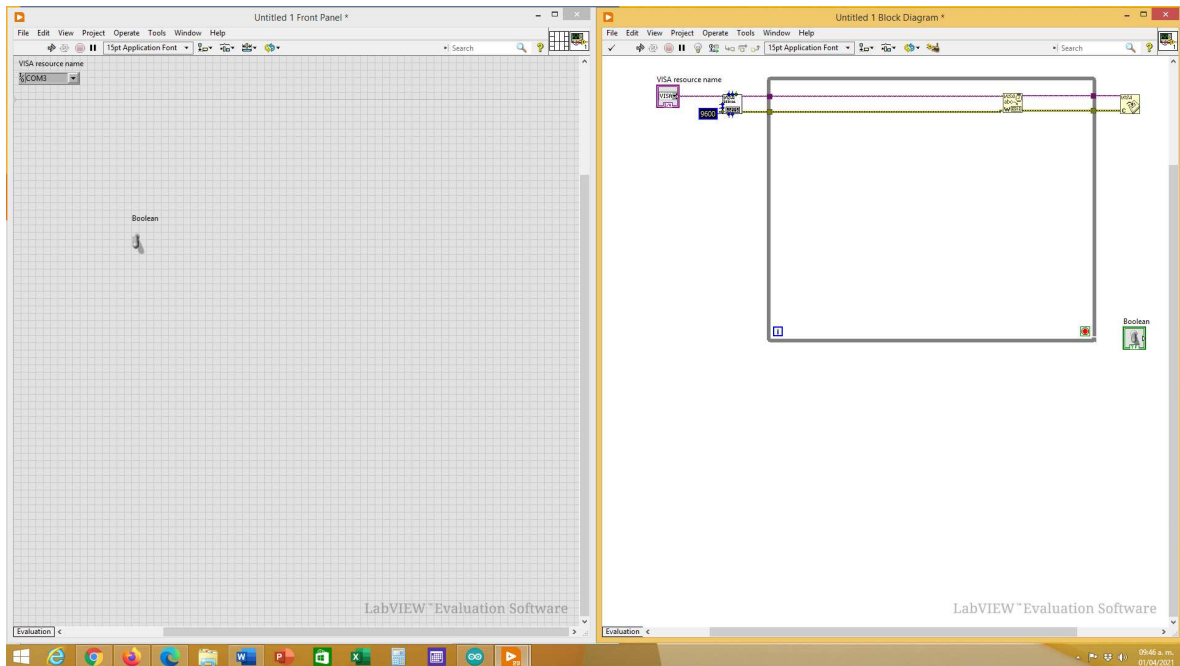
Como se observa, en la ventana de Panel Frontal, aparece un control. En este control debemos elegir el Puerto Serial de comunicación que corresponda con el Arduino



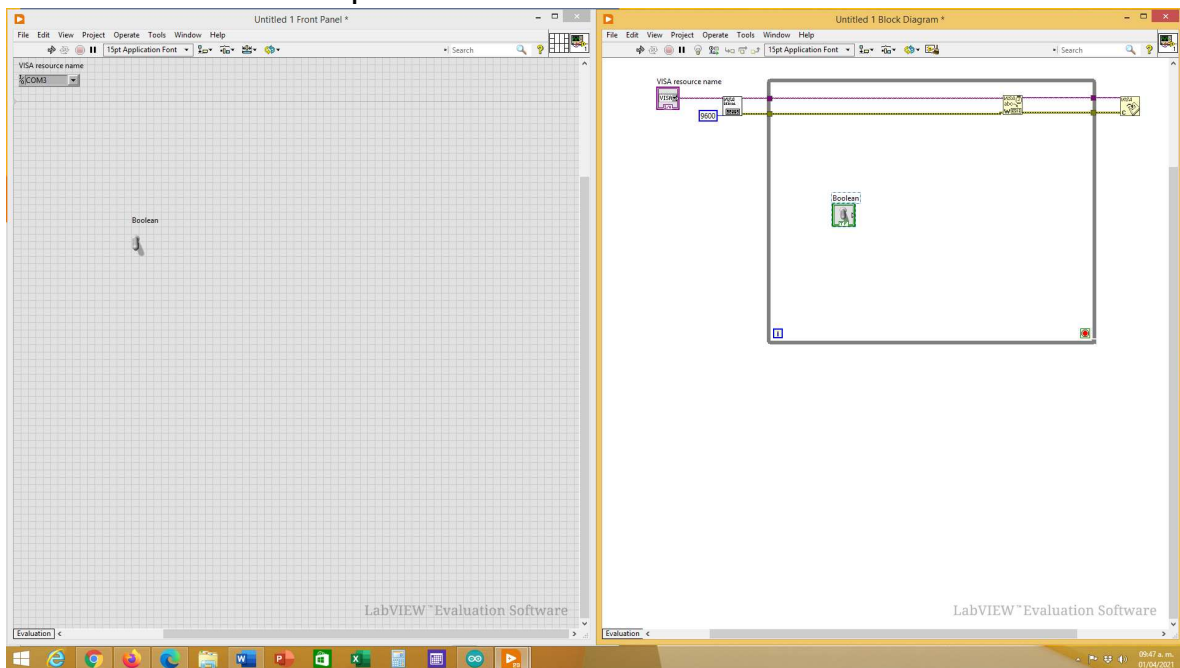
Despues en la ventana de Diagrama de Bloques, damos clic en el segundo conector (baud rate 9600) y elegimos Create Constant



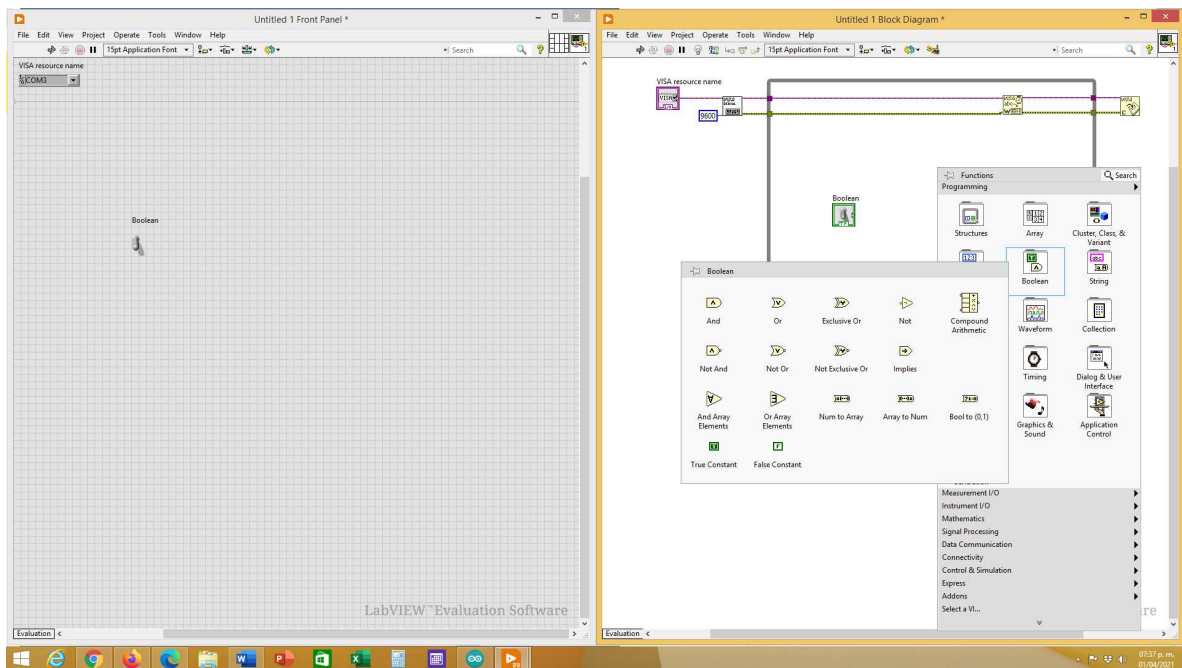
Por default nos aparece el valor de 9600. Este valor dejamos intacto para no tener problemas con la velocidad y sincronización de los datos



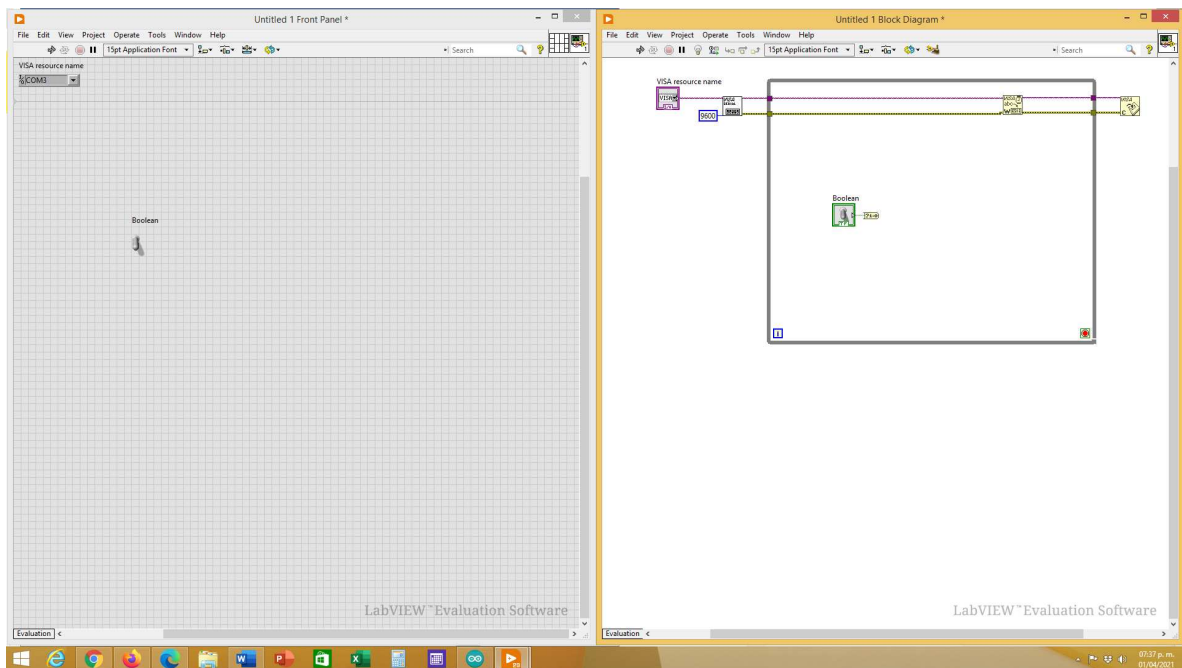
Arrastramos el interruptor hacia dentro del control While.



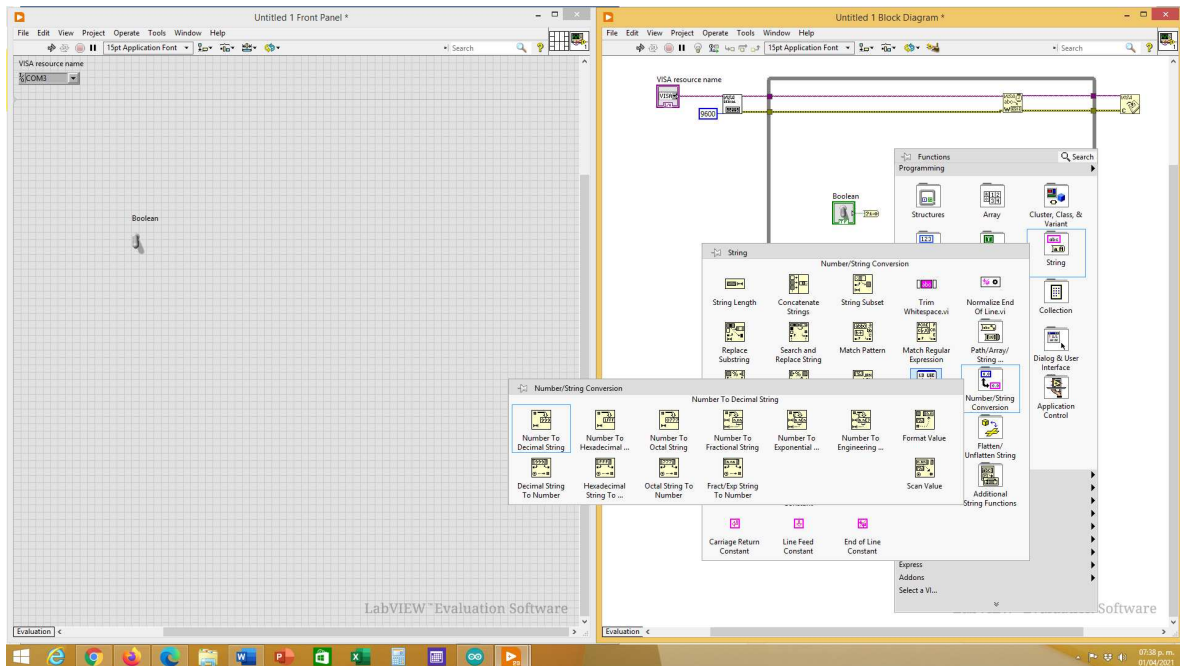
Clic con boton derecho y en Boolean, elegimos Bool to(0,1)



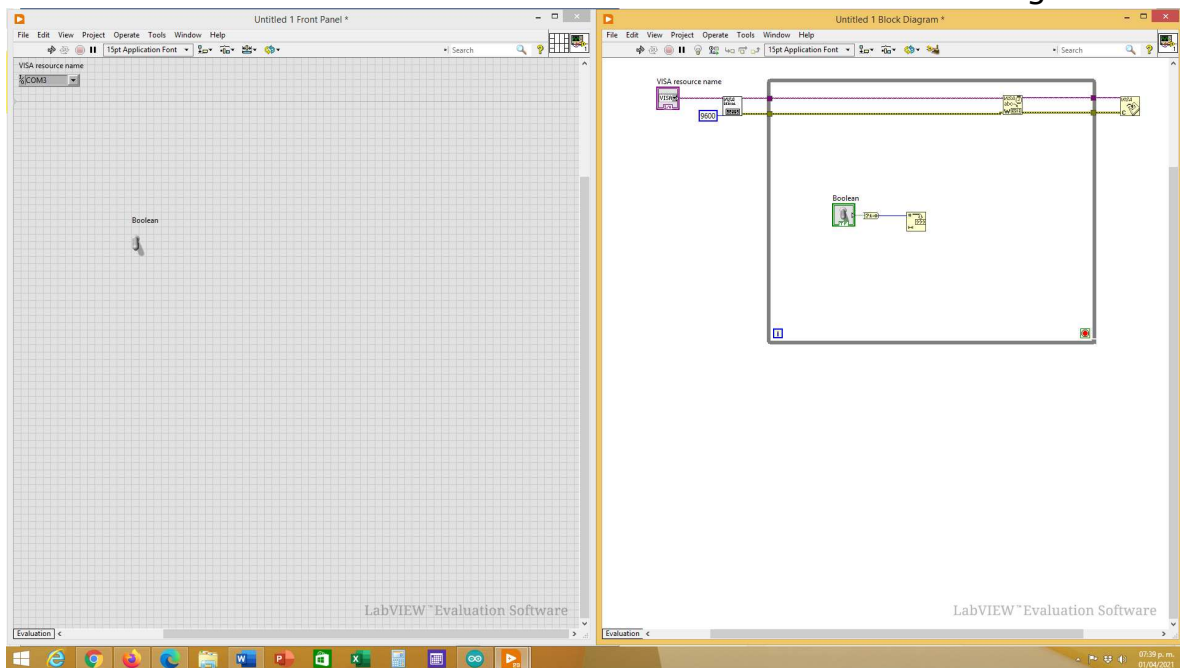
Conectamos este control acercandolo al icono del switch



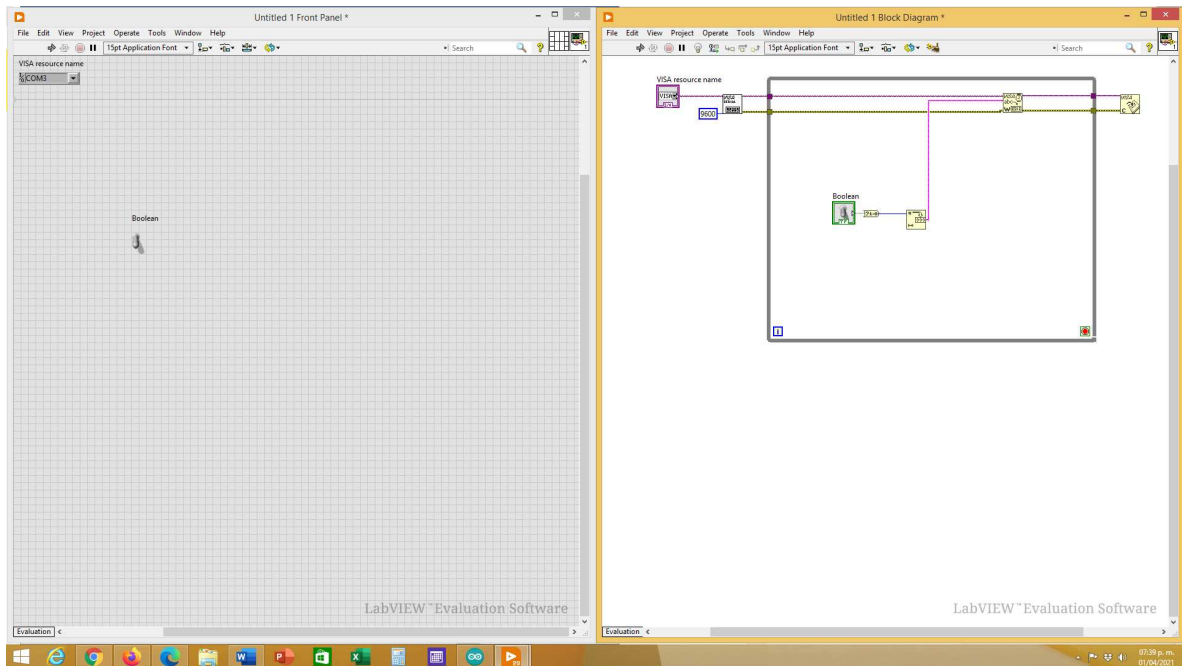
Continuamos ingresando el icono Number to Decimal String dando clic en el menu String, y posteriormente en la opción Number/String Conversion



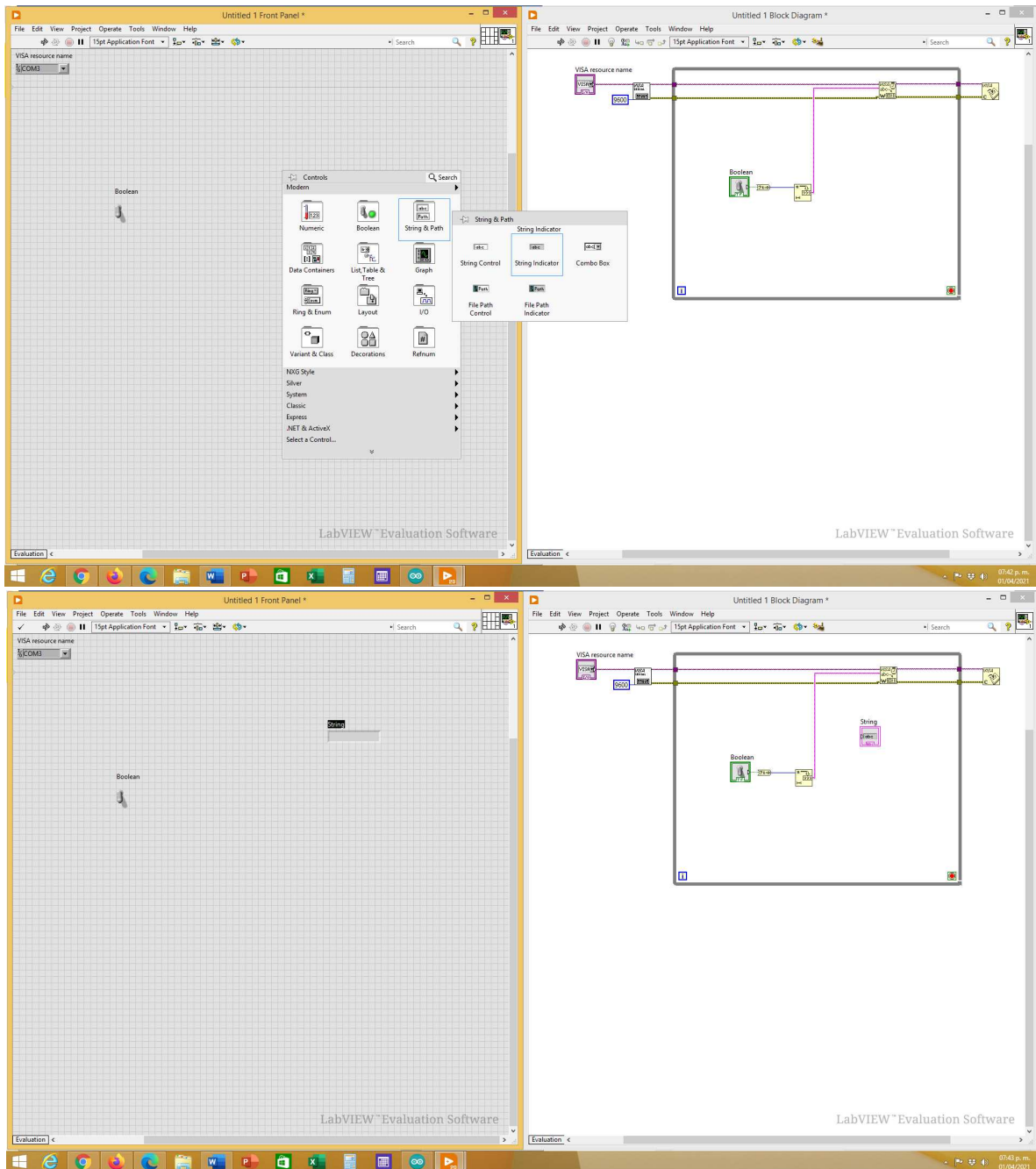
Conectamos este ultimo icono al anterior como se muestra en la figura



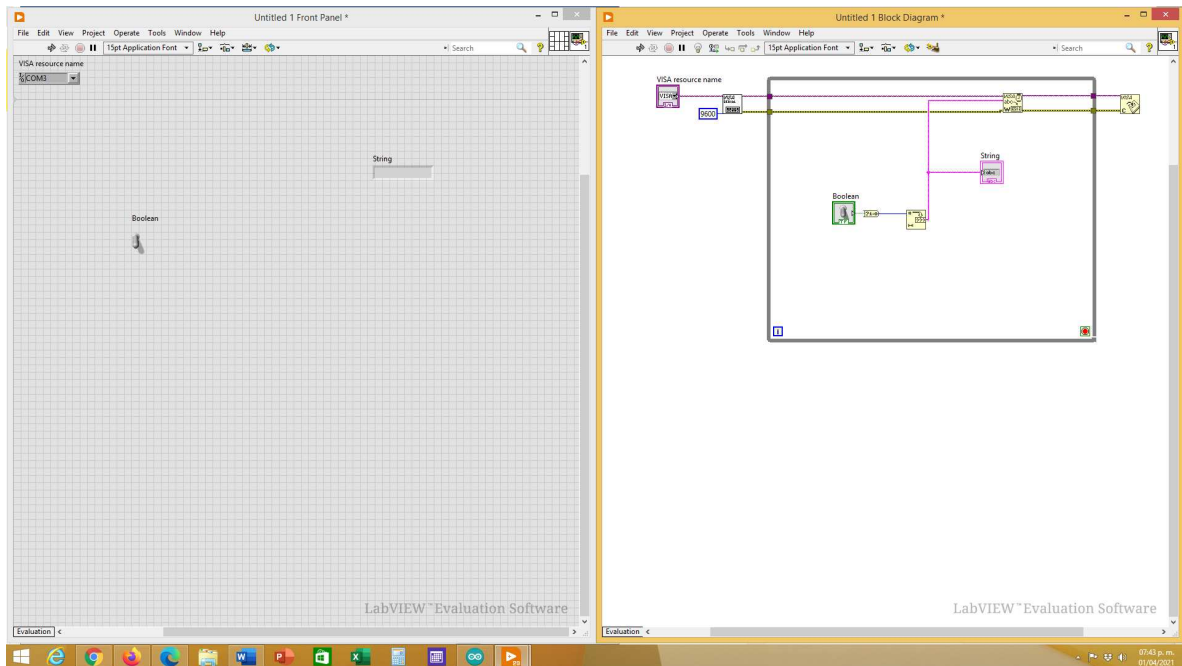
De esta manera, el valor booleano, se convierte en cadena para poder viajar por el puerto serial. Conectamos esta salida a la entrada ABC del control WRITE



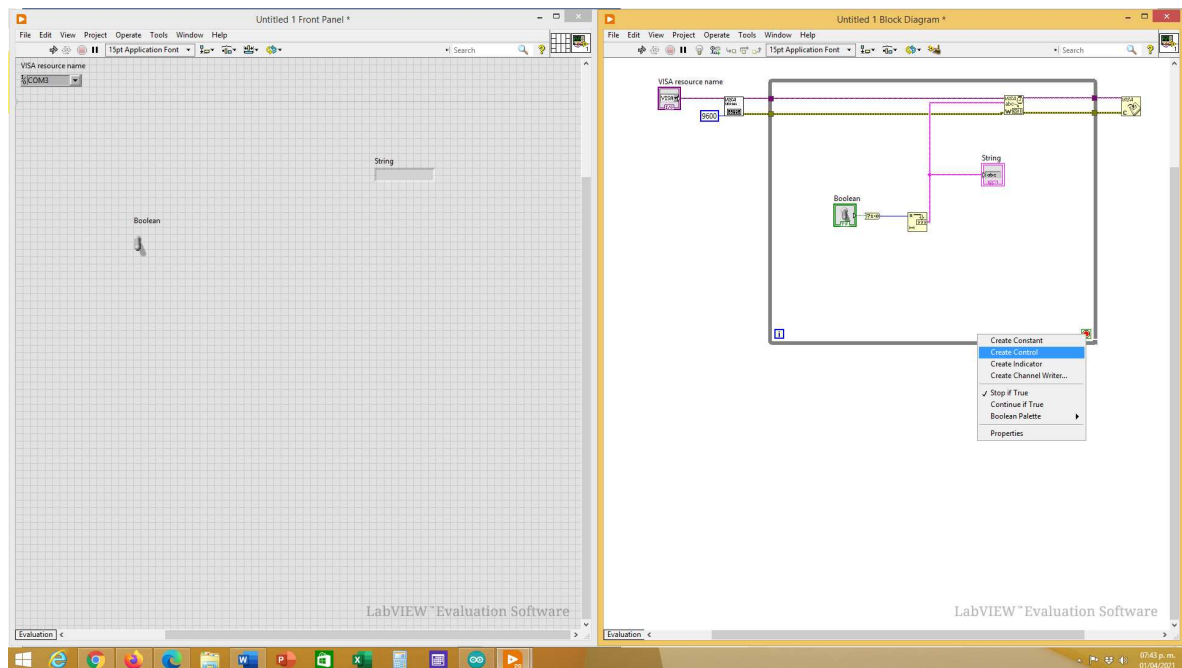
Metemos un control para monitorear el valor que se esta enviando. No es necesario pero sirve para testear



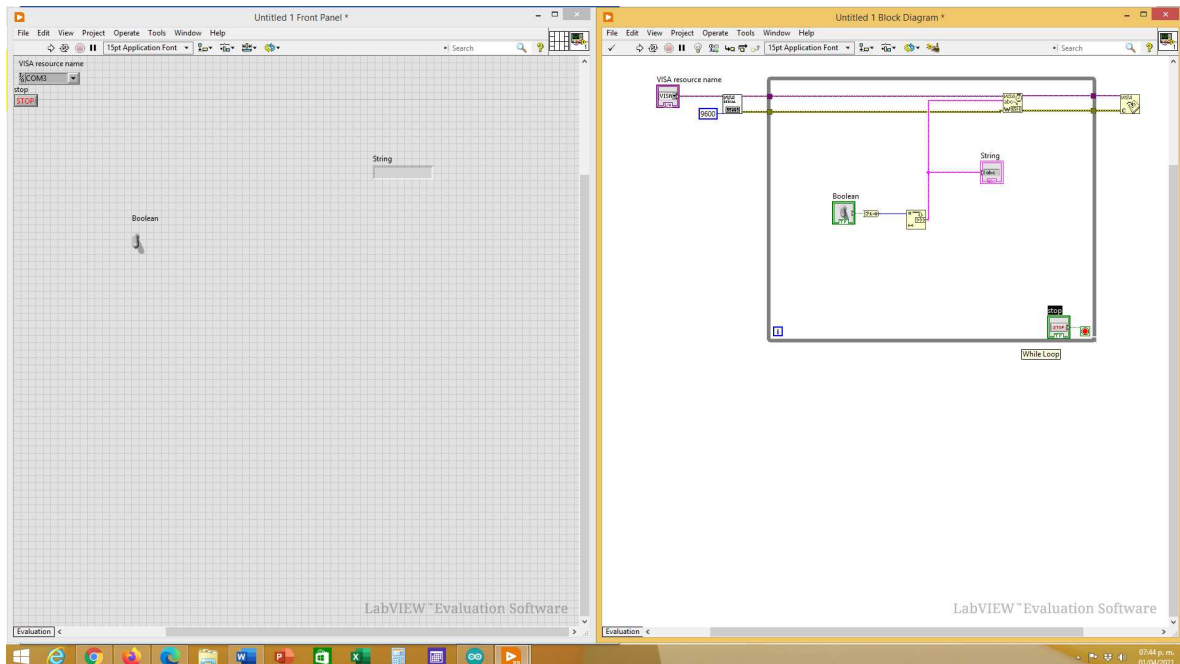
Conectamos este control a la misma línea de salida que va al Write. Si se envía 0 o 1 ahí en esa pantalla se visualizará.



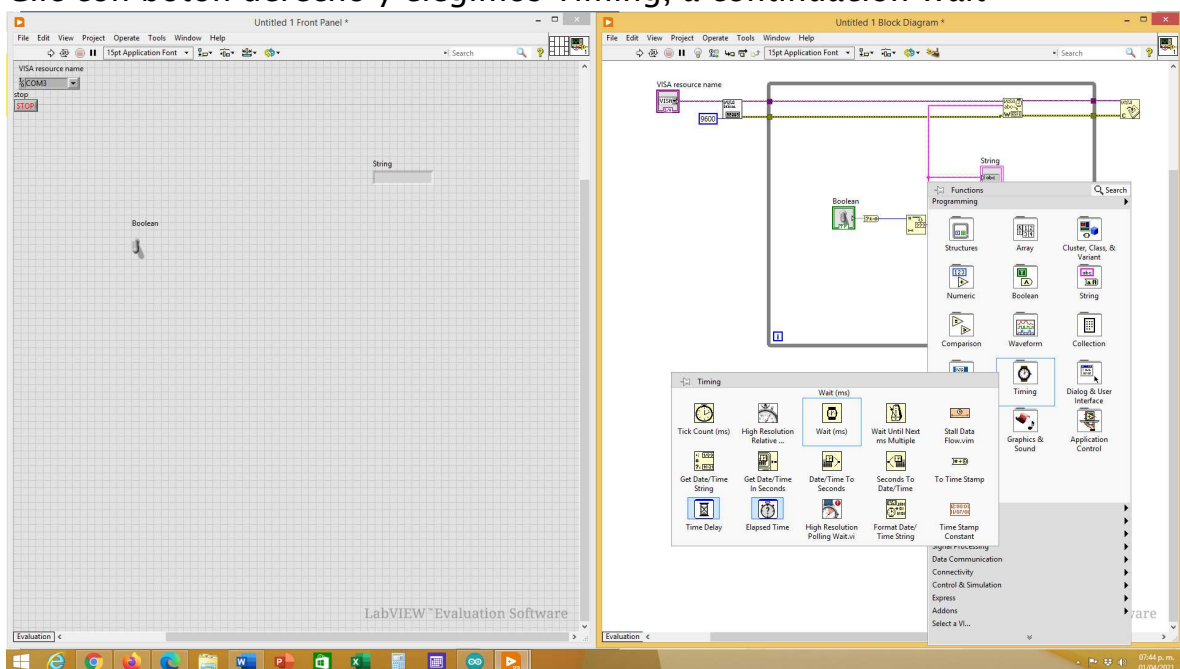
Por ultimo en el iconito rojo de la derecha abajo, damos clic para agregar un control



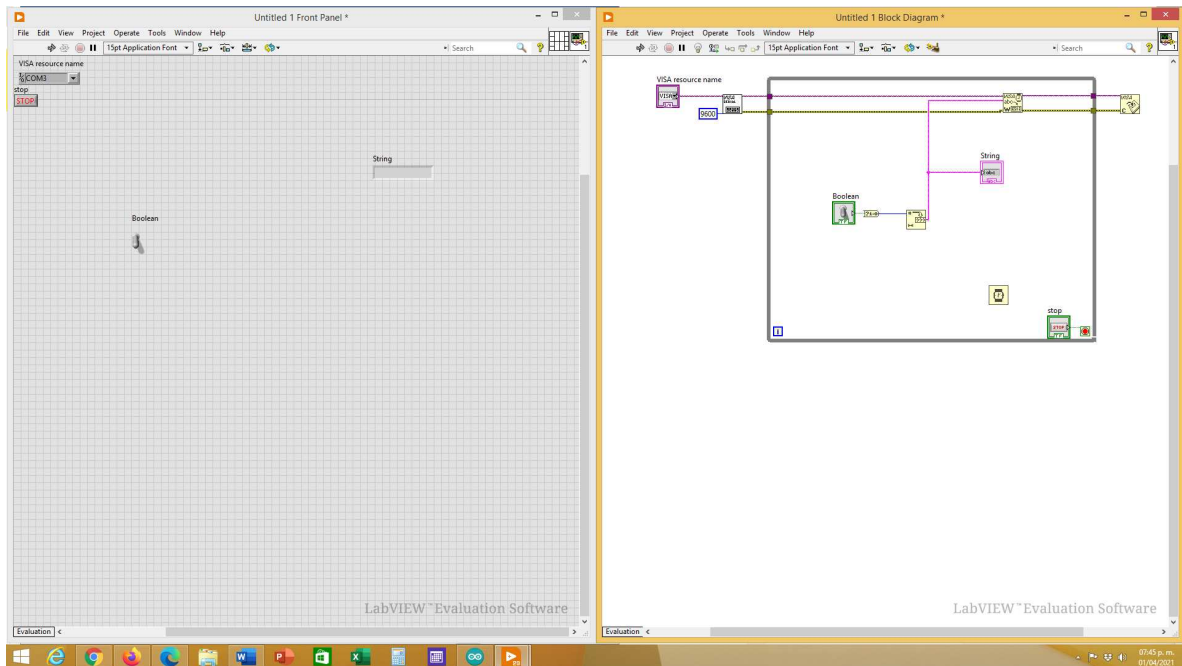
De lado del Panel frontal tambien aparecera el boton STOP que es como un paro de emergencia que se tiene que agregar en todas las apps



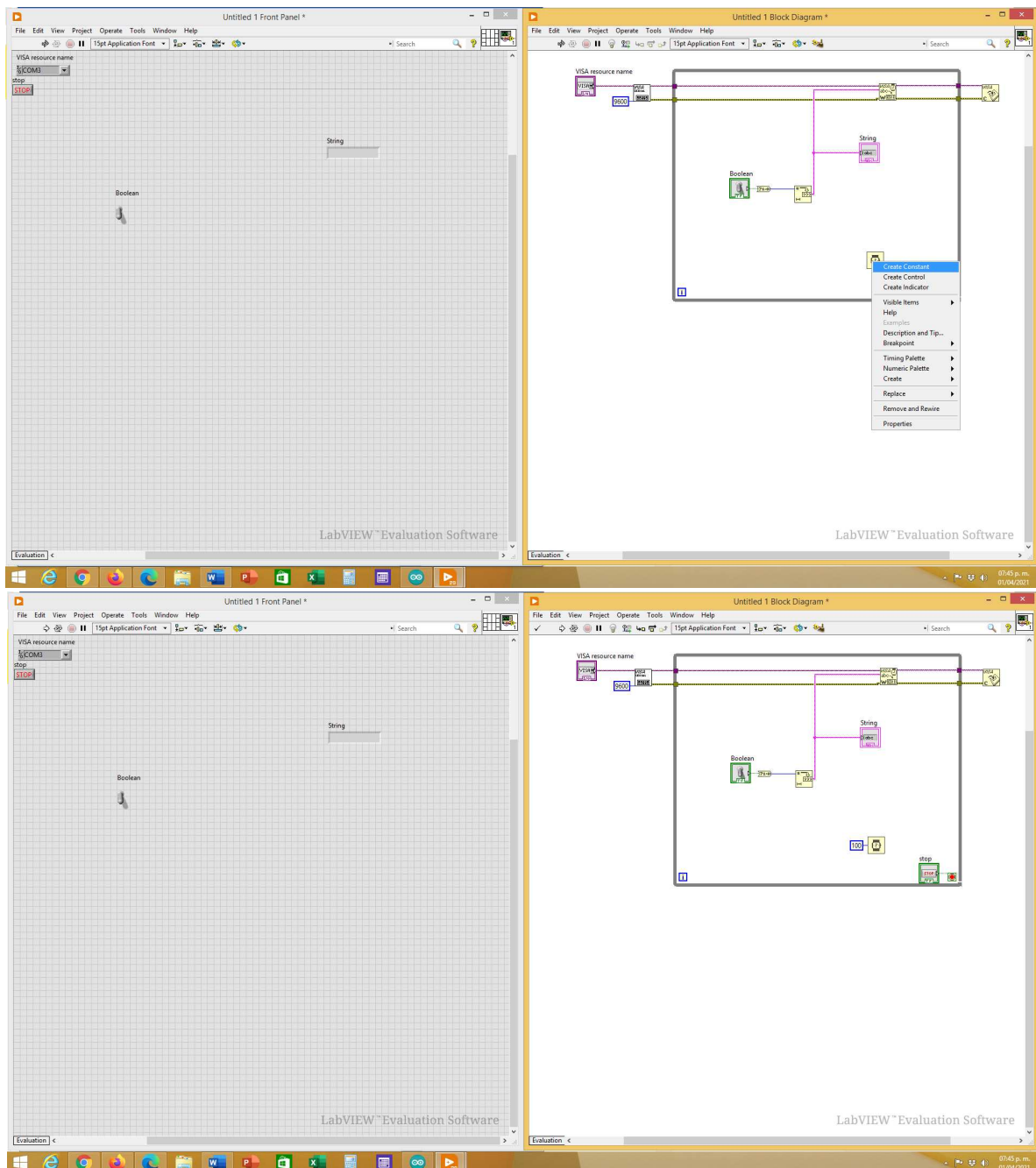
Para sincronizar este programa con Arduino, necesitamos agregar una señal de reloj a 100ms
Clic con boton derecho y elegimos Timing, a continuacion Wait



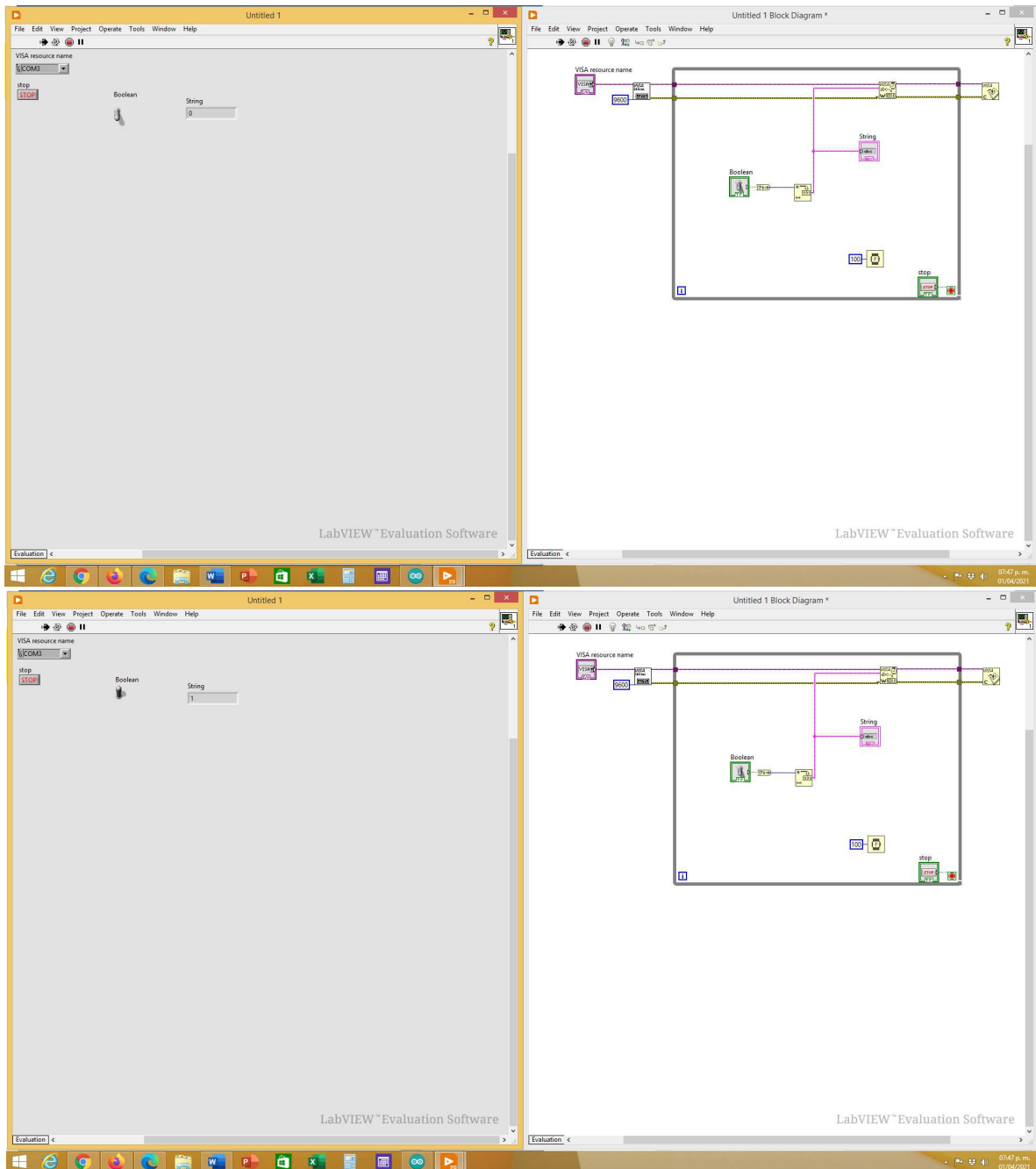
Quedaria asi como en la pantalla

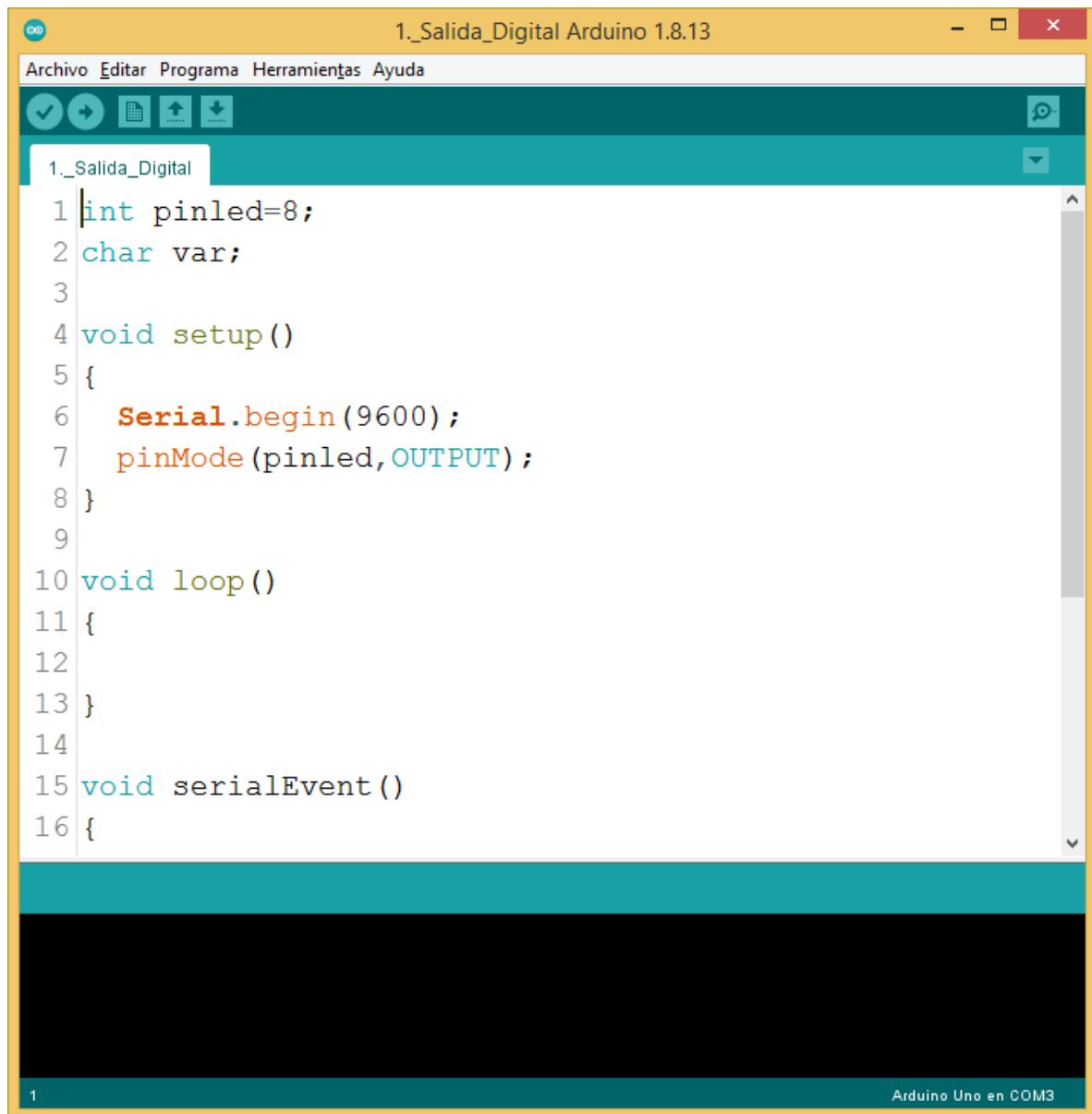


Clic nuevamente sobre el mismo control y elegimos crear constante, ahí colocamos 100



Ejecutamos y si no hay error, podemos continuar con el programa de Arduino

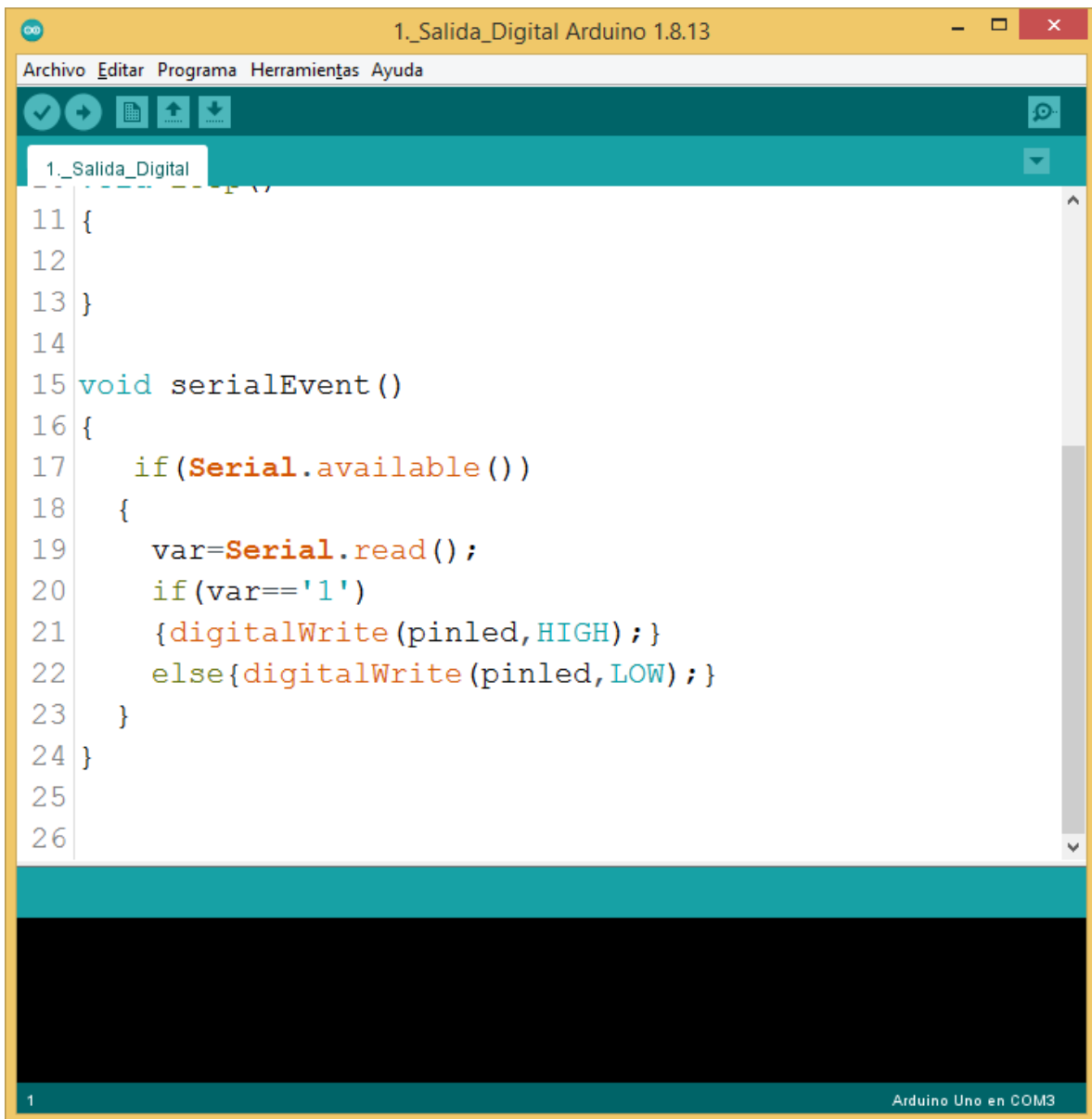




The image shows the Arduino IDE 1.8.13 window. The title bar reads "1_Salida_Digital Arduino 1.8.13". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for opening, saving, compiling, uploading, and erasing. The file explorer shows a single file named "1_Salida_Digital". The code editor contains the following C++ code:

```
1 int pinled=8;
2 char var;
3
4 void setup()
5 {
6     Serial.begin(9600);
7     pinMode(pinled, OUTPUT);
8 }
9
10 void loop()
11 {
12
13 }
14
15 void serialEvent()
16 {
```

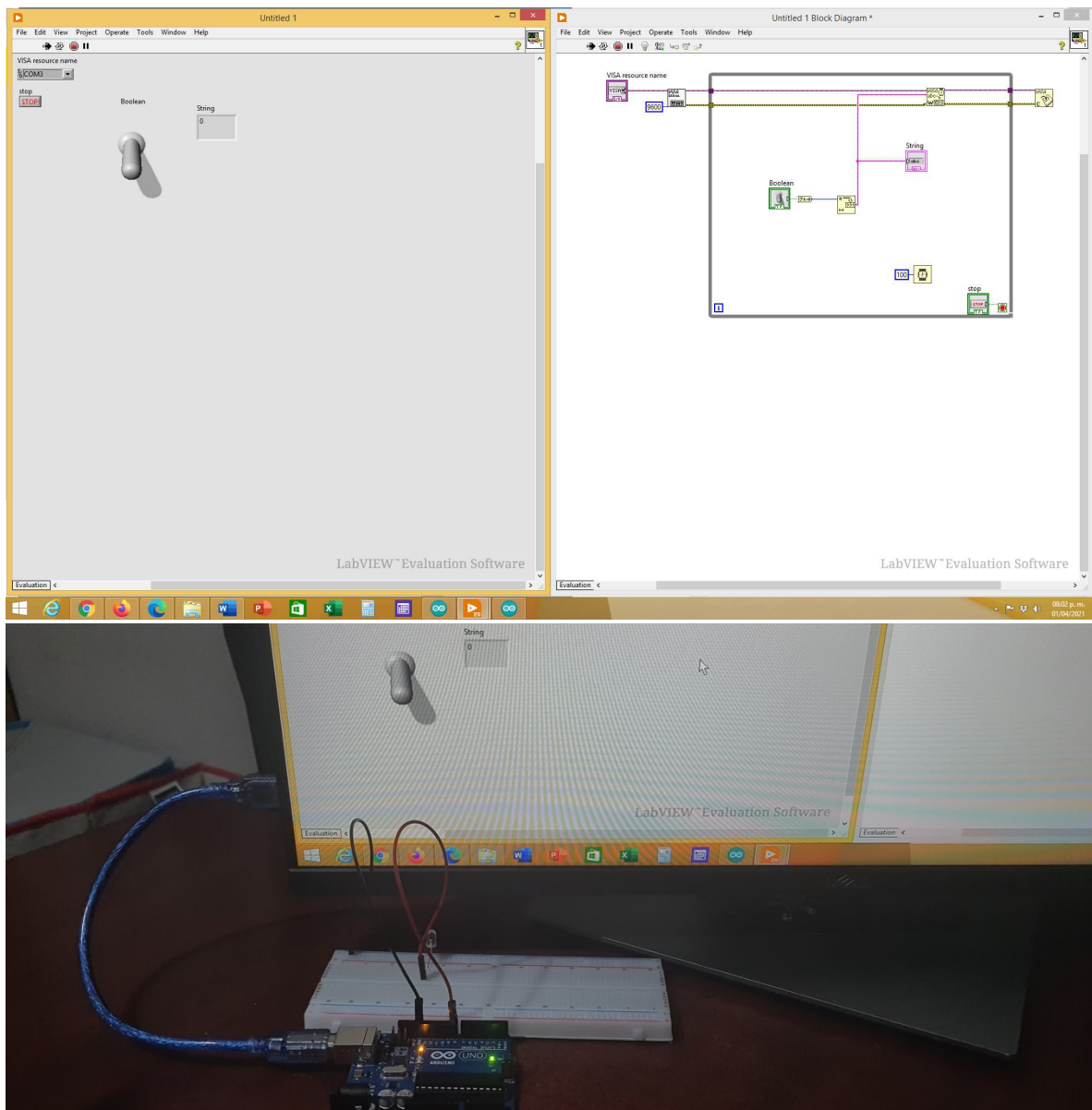
Below the code editor is a black area for the serial monitor. At the bottom left, the line number "1" is displayed. At the bottom right, the text "Arduino Uno en COM3" is shown.

The image shows the Arduino IDE interface with a file named "1_Salida_Digital". The code is written in C++ and is designed to control an LED based on serial input. The code includes a setup function (partially visible) and a serialEvent function. The serialEvent function checks if data is available on the serial port. If the received character is '1', it sets the LED pin (pinled) to HIGH, turning the LED on. If the character is anything else, it sets the pin to LOW, turning the LED off. The IDE window shows line numbers from 11 to 26. The status bar at the bottom indicates "1" and "Arduino Uno en COM3".

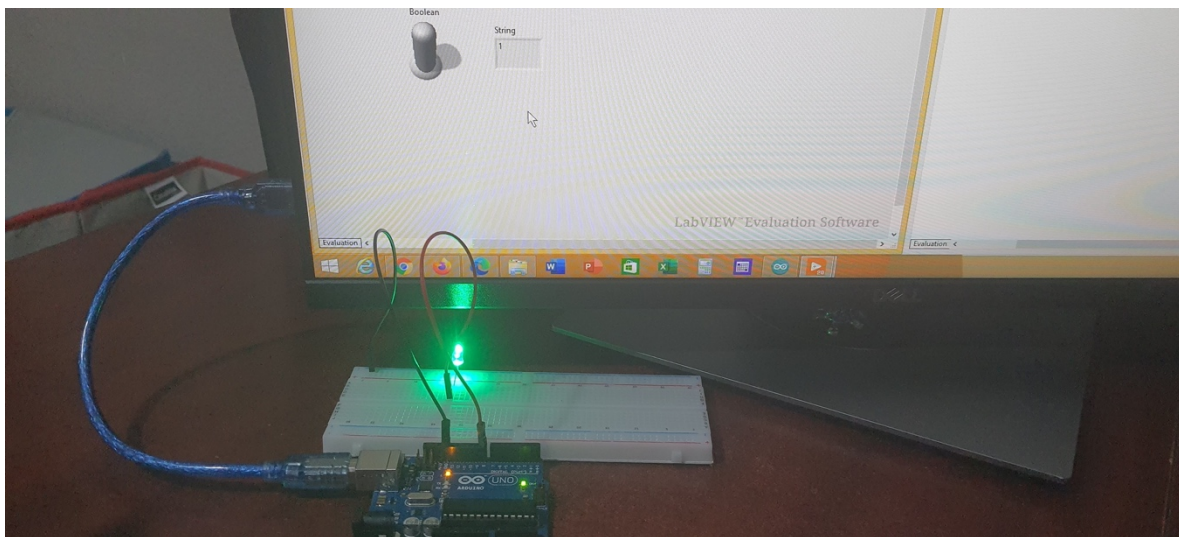
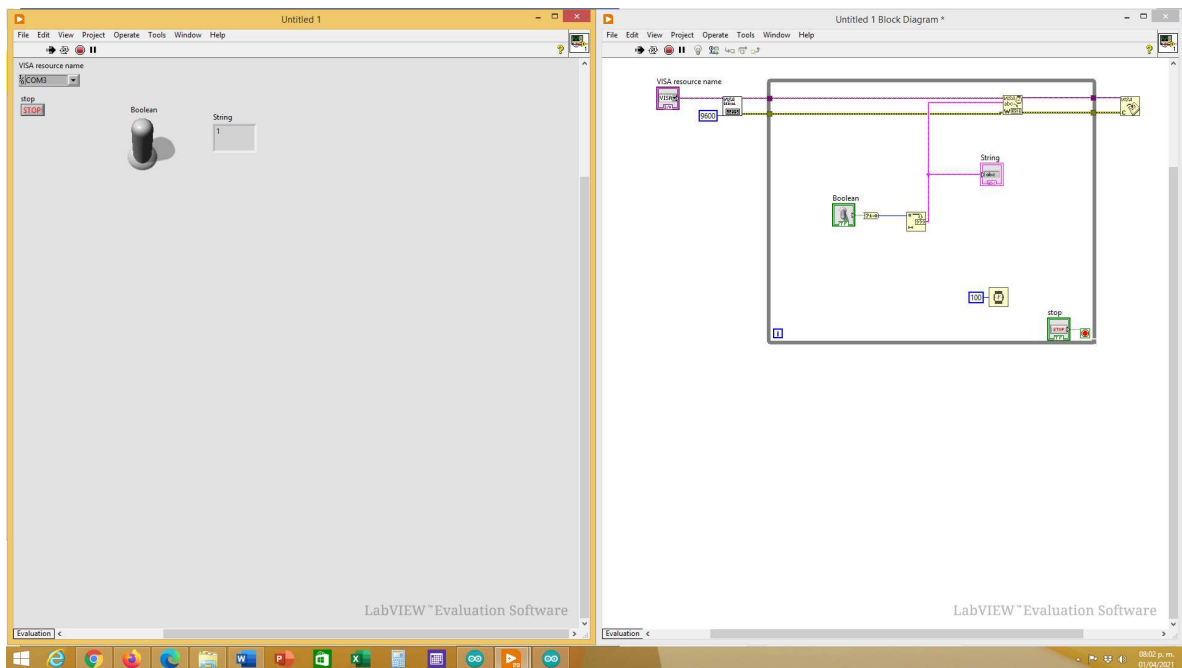
```
11 {  
12  
13 }  
14  
15 void serialEvent()  
16 {  
17     if(Serial.available())  
18     {  
19         var=Serial.read();  
20         if(var=='1')  
21             {digitalWrite(pinled,HIGH);}  
22         else{digitalWrite(pinled,LOW);}  
23     }  
24 }  
25  
26
```

Se carga el programa de Arduino a la placa, que consiste en detectar si en el puerto serial hay algun dato disponible y si este es el carácter 1, encendera el led conectado en el pin 8. Si no es el valor de 1, entonces apagara el led.

Ejecutamos y cuando la palanca este abajo, el led fisico debera estar apagado



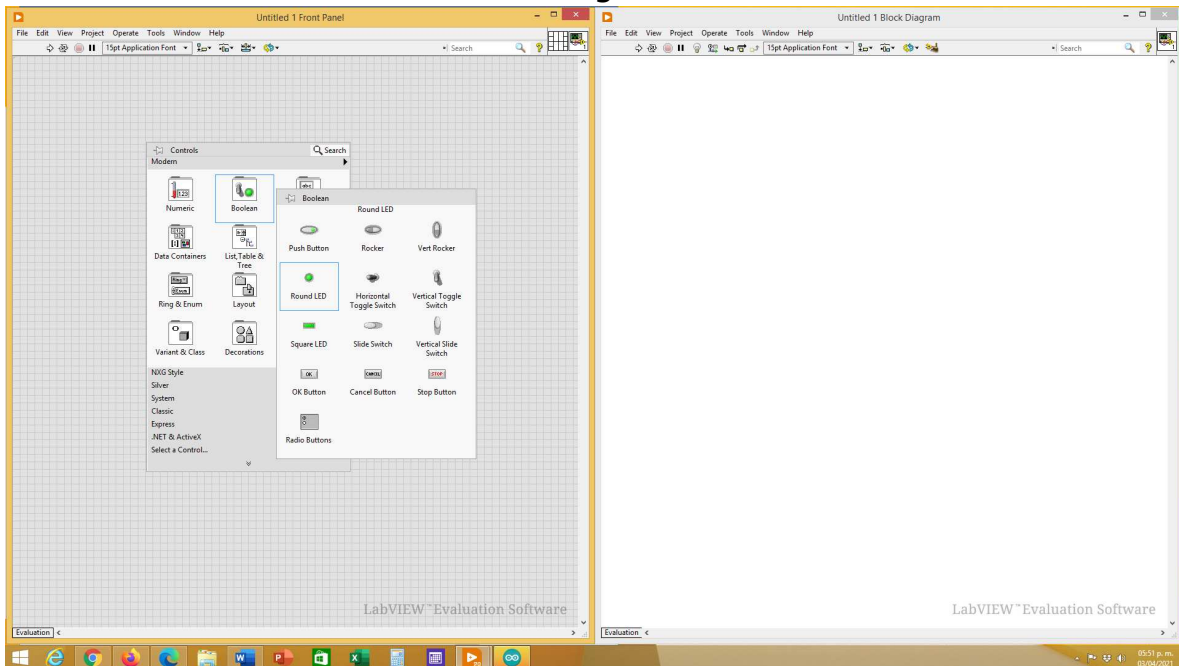
Si movemos la palanca hacia arriba, el led debera encender



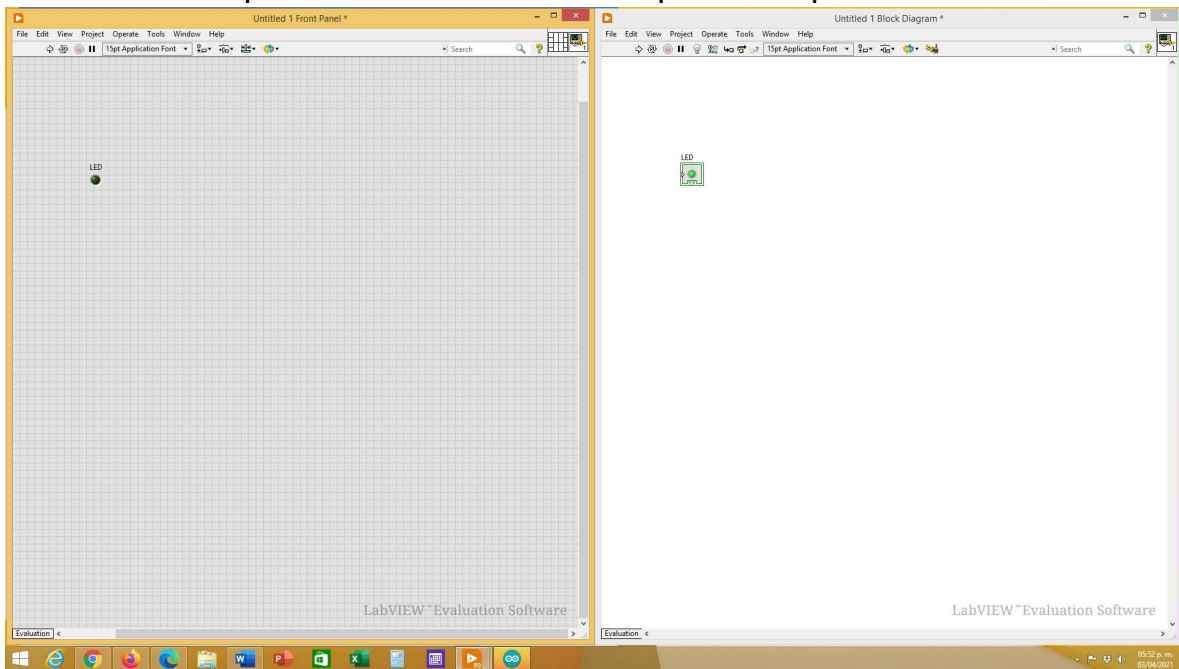
PRACTICA 12: Lectura de un boton desde Arduino, monitoreado por Labview

Abrir Labview y dar click en Nuevo VI

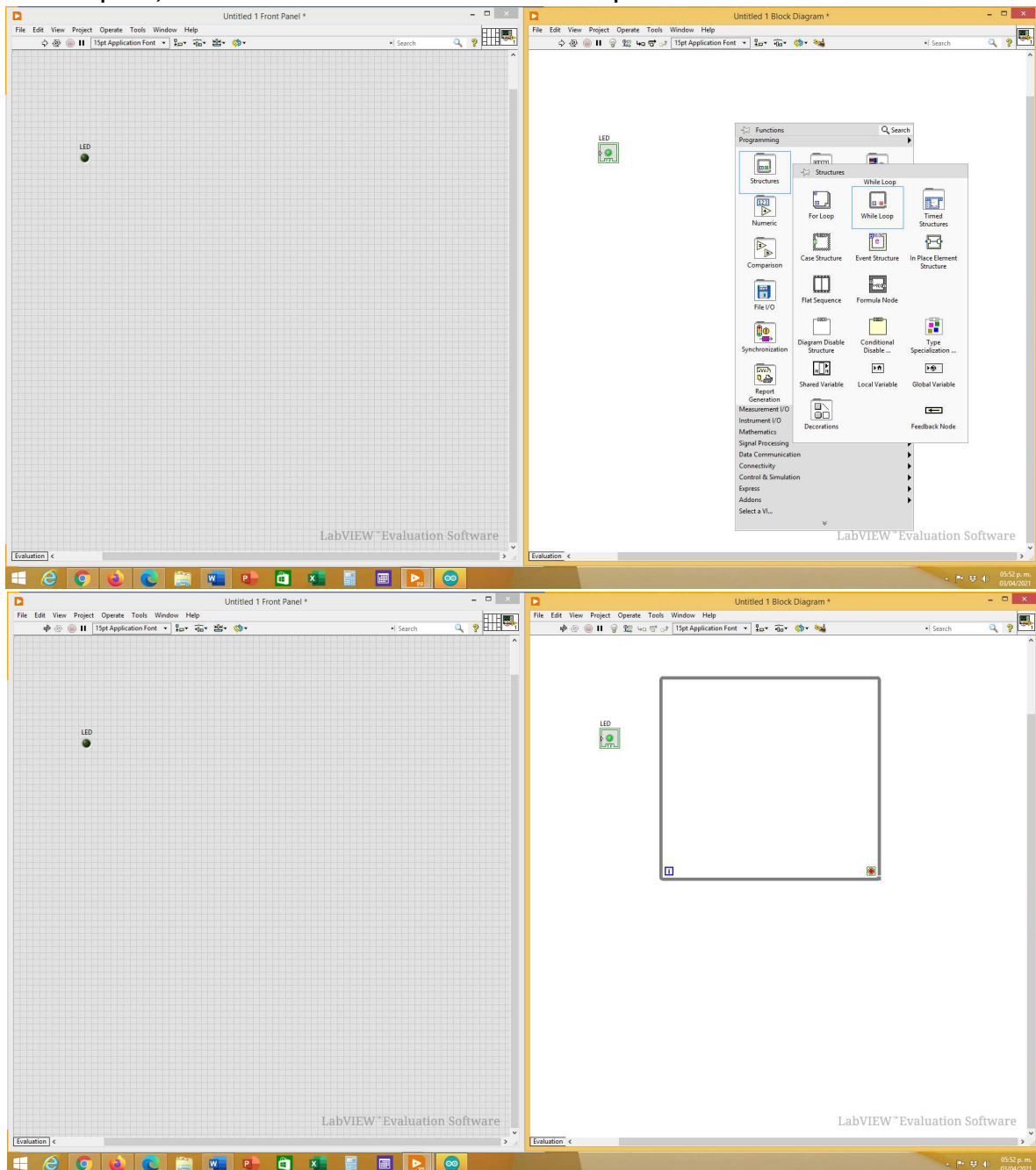
En la parte de Panel Frontal agregamos un led usando los controles Booleanos como se muestra en la imagen.



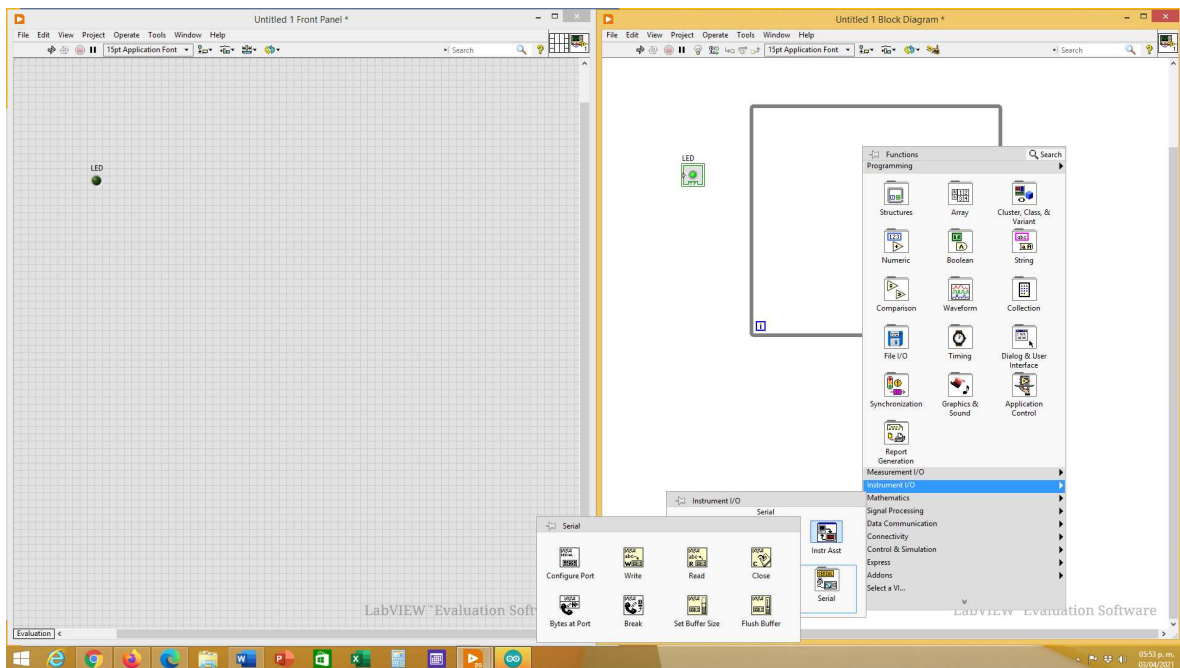
De lado de bloques encontraremos el bloque correspondiente



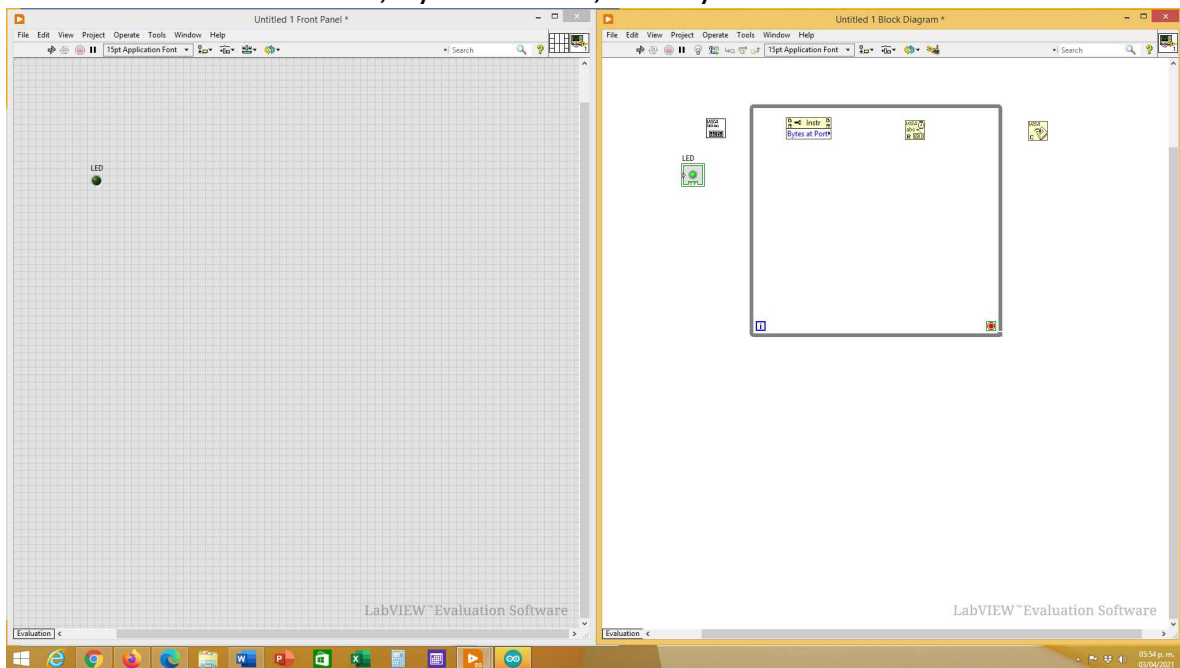
En Bloques, metemos la estructura While loop



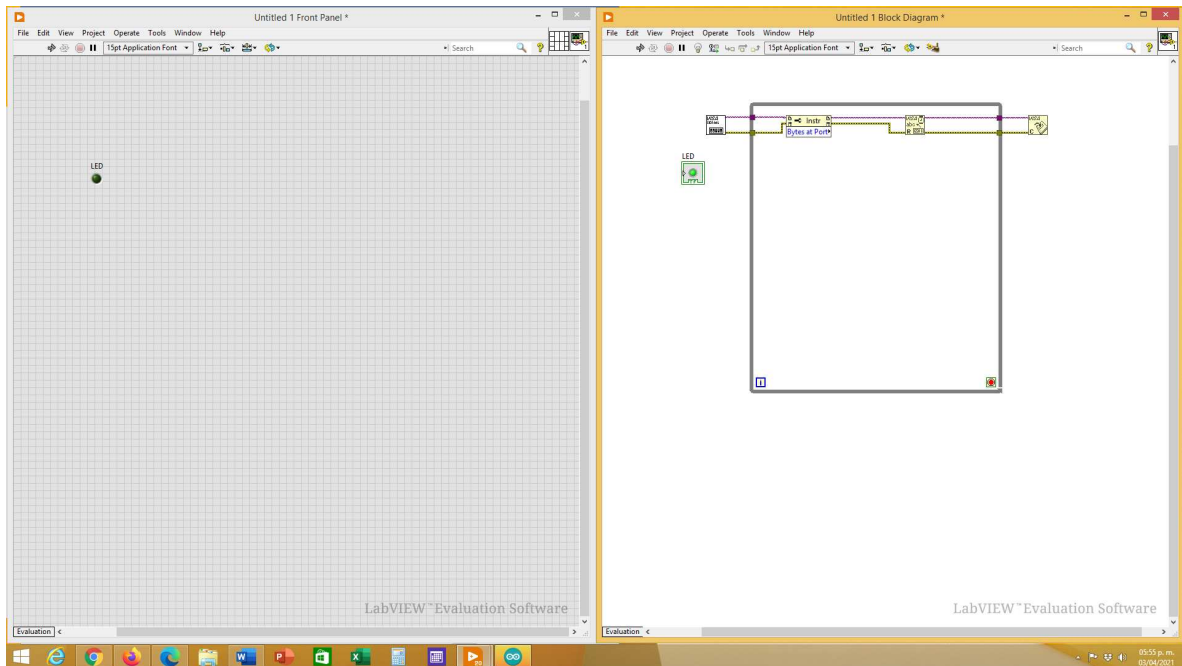
Posteriormente, integraremos los bloques para comunicación Serial, estos los sacamos del menu Instrument I/O



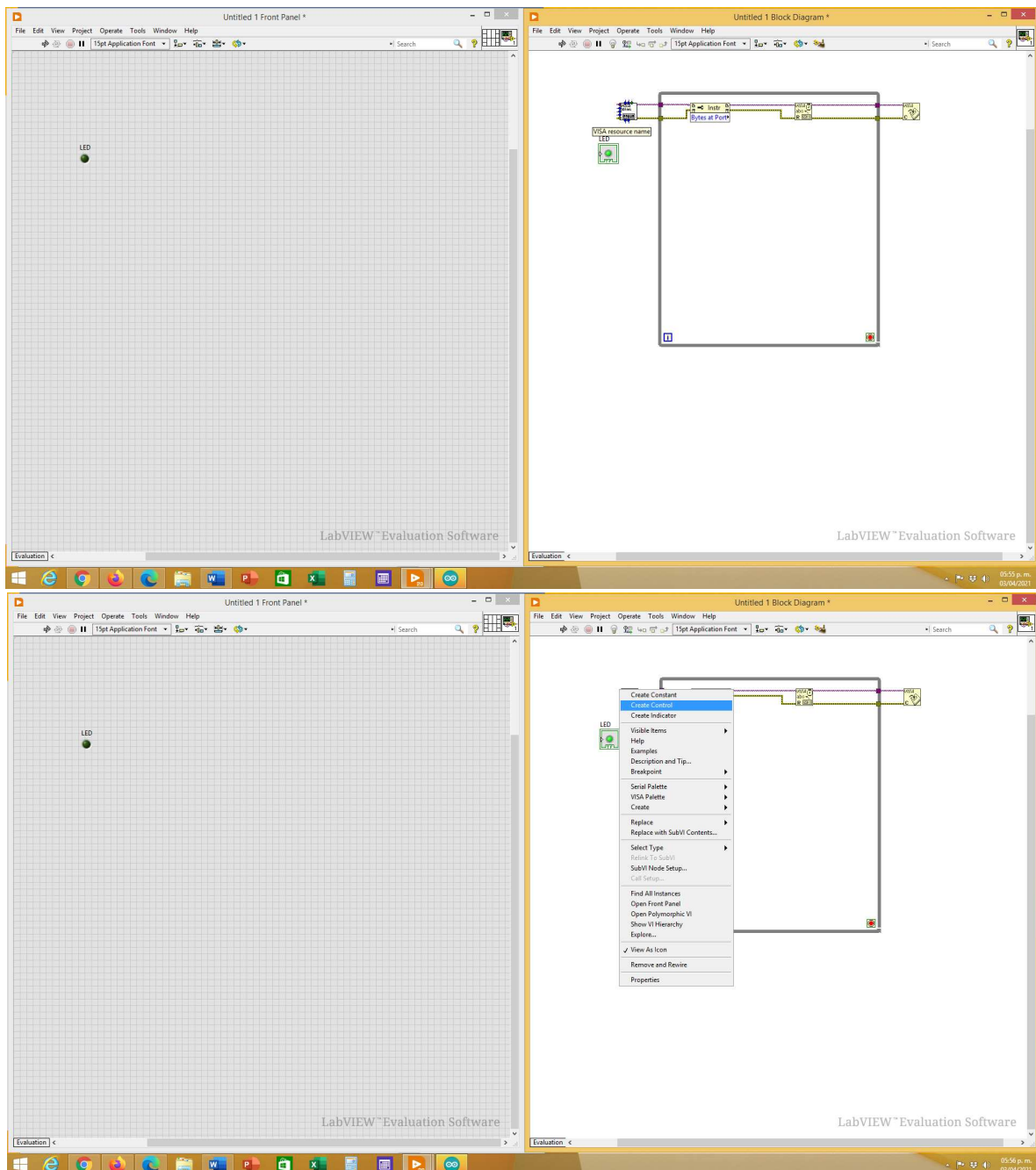
Necesitamos VISA Serial, Bytes at Port, Read y Close.



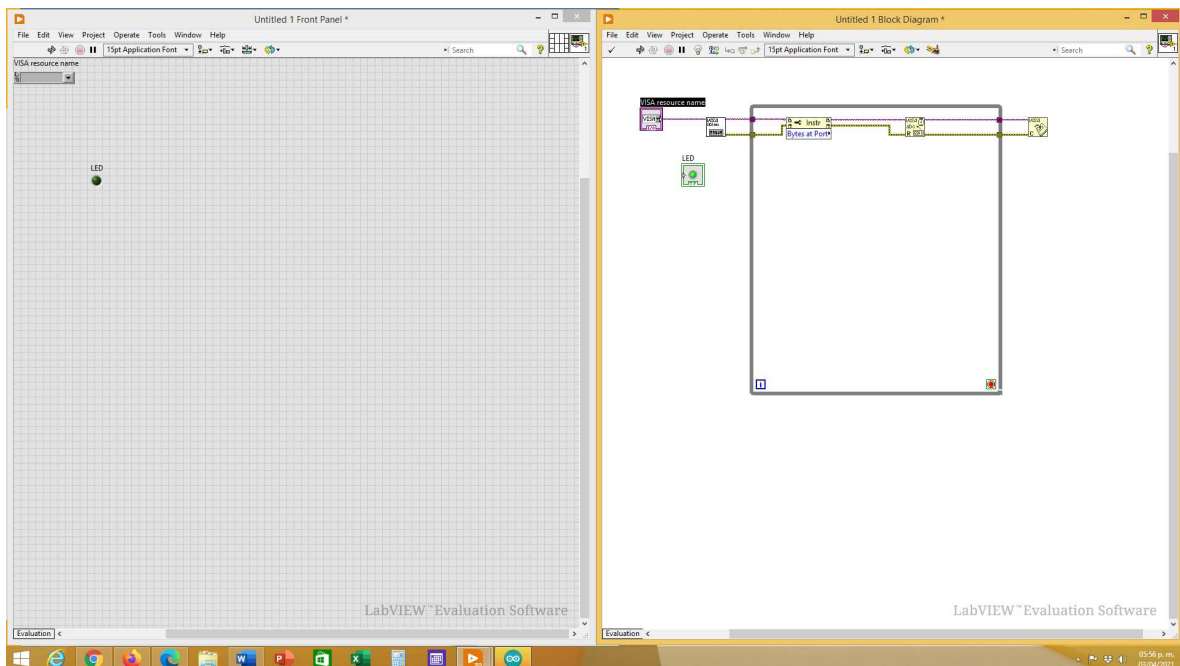
Realizamos las conexiones de entradas y salidas como se muestra en la siguiente imagen, como se observa además, se crea un tunel con el marco del while loop



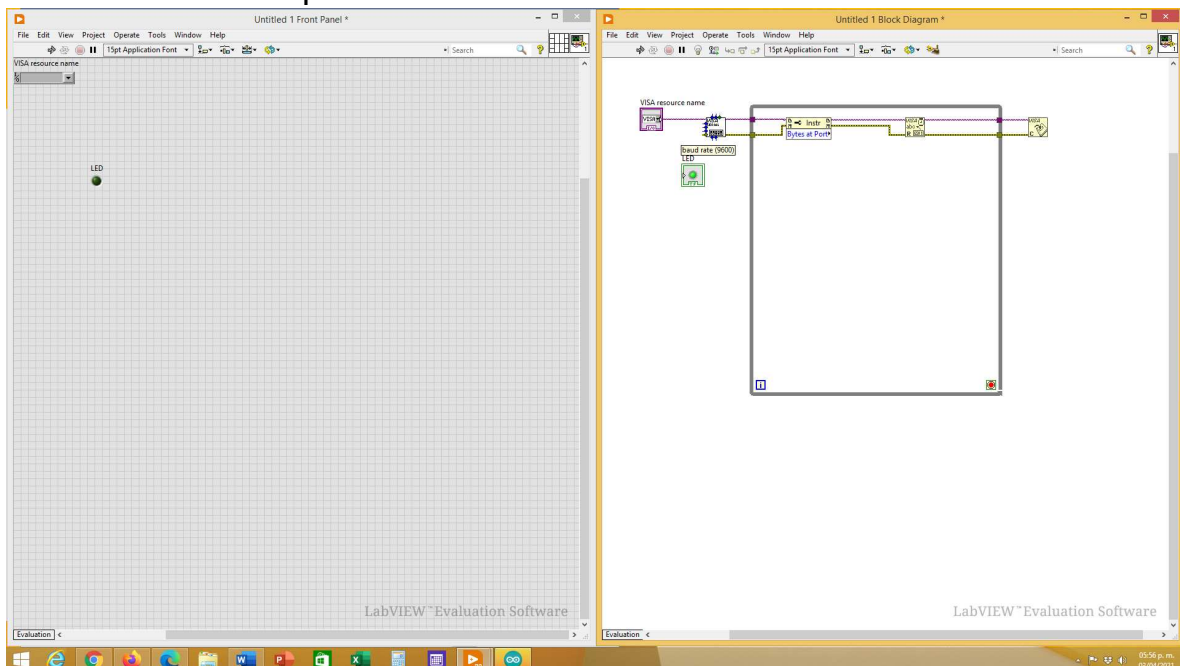
En el bloque de VISA SERIAL, damos clic en el primer conector y elegimos CREAR Control



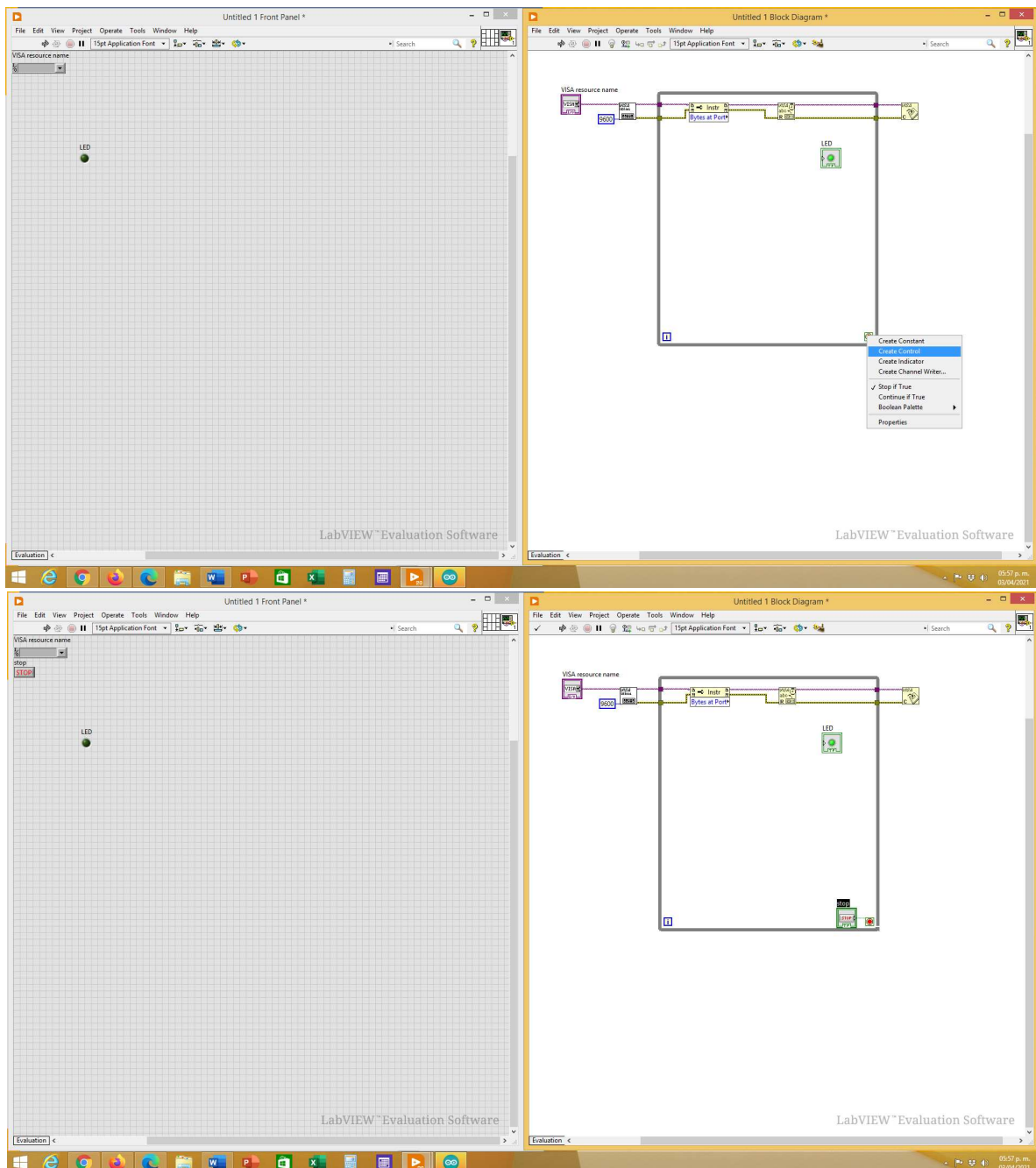
Aparecera en la pantalla un nuevo bloque, al igual que en la pantalla del Panel Frontal. En este podremos elegir el Puerto de comunicación con Arduino que debera ser el mismo COM



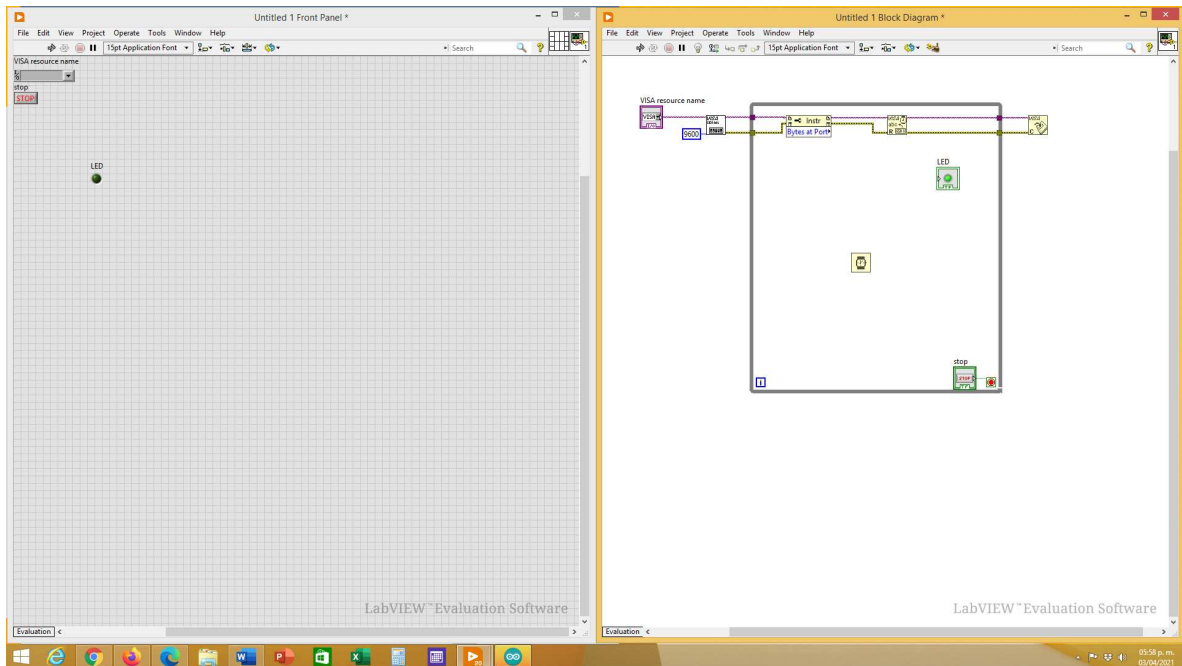
En el mismo bloque, pero en el segundo conector, elegimos baud rate y seleccionamos la opción de CREAR Constante.



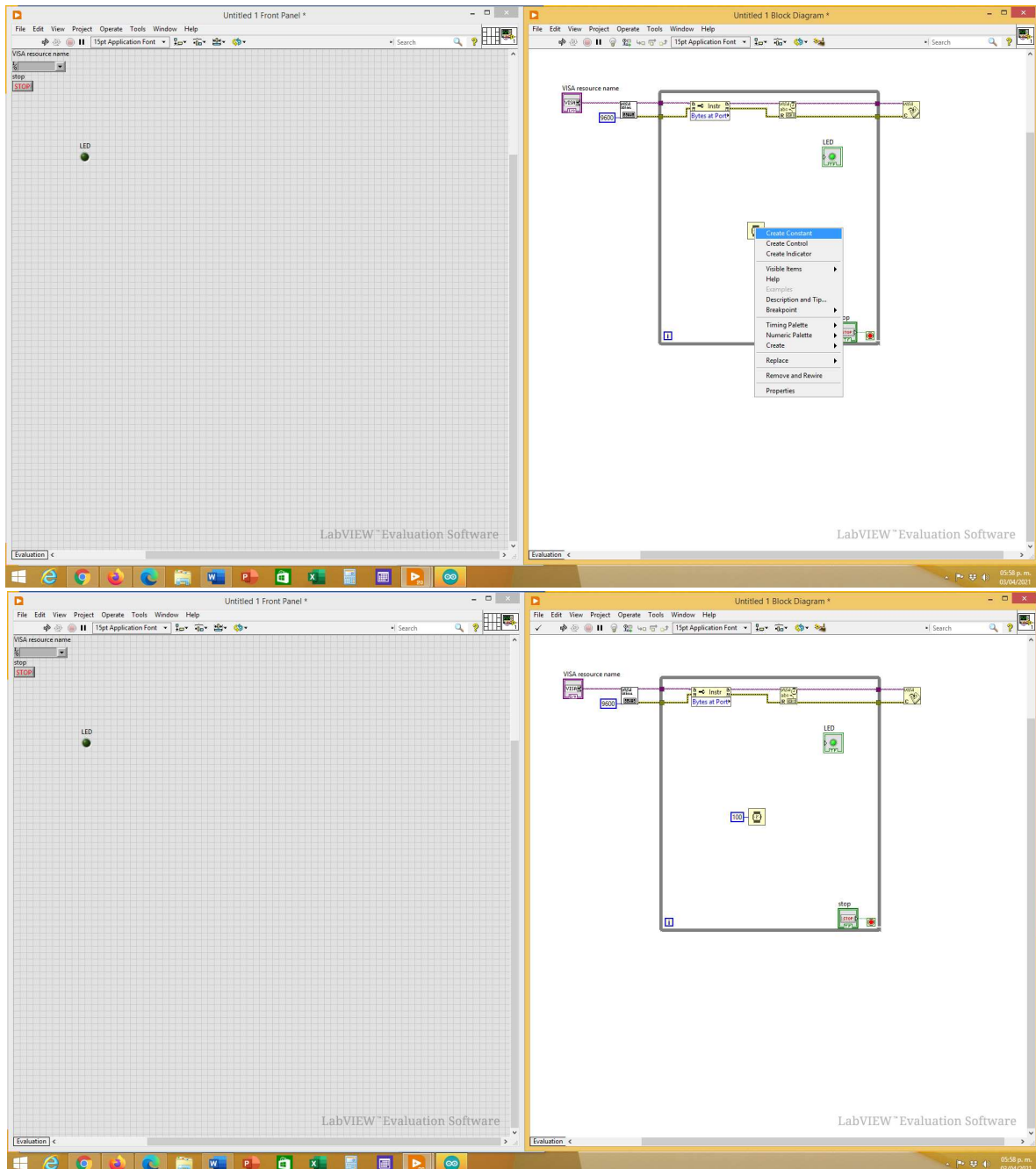
El valor que debemos colocar es de 9600 que sera el mismo que emplearemos con Arduino.



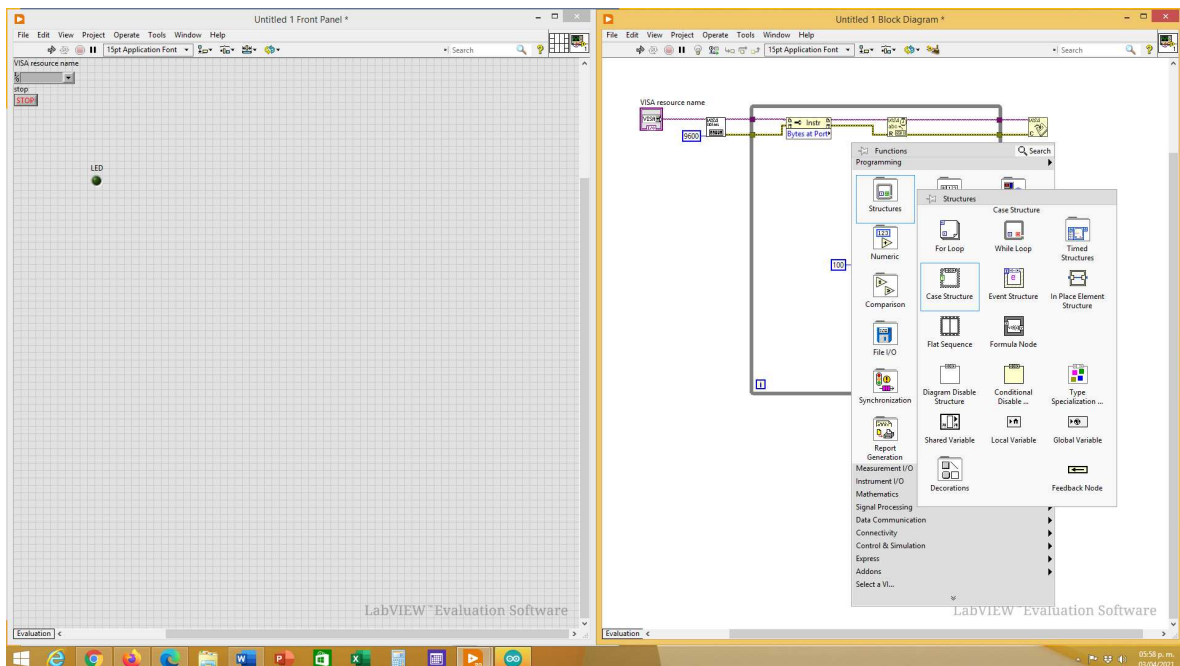
La señal debe estar sincronizada con el Arduino, para lo cual, colocaremos un Reloj a 100ms, esto lo haremos ingresando un bloque de Timing



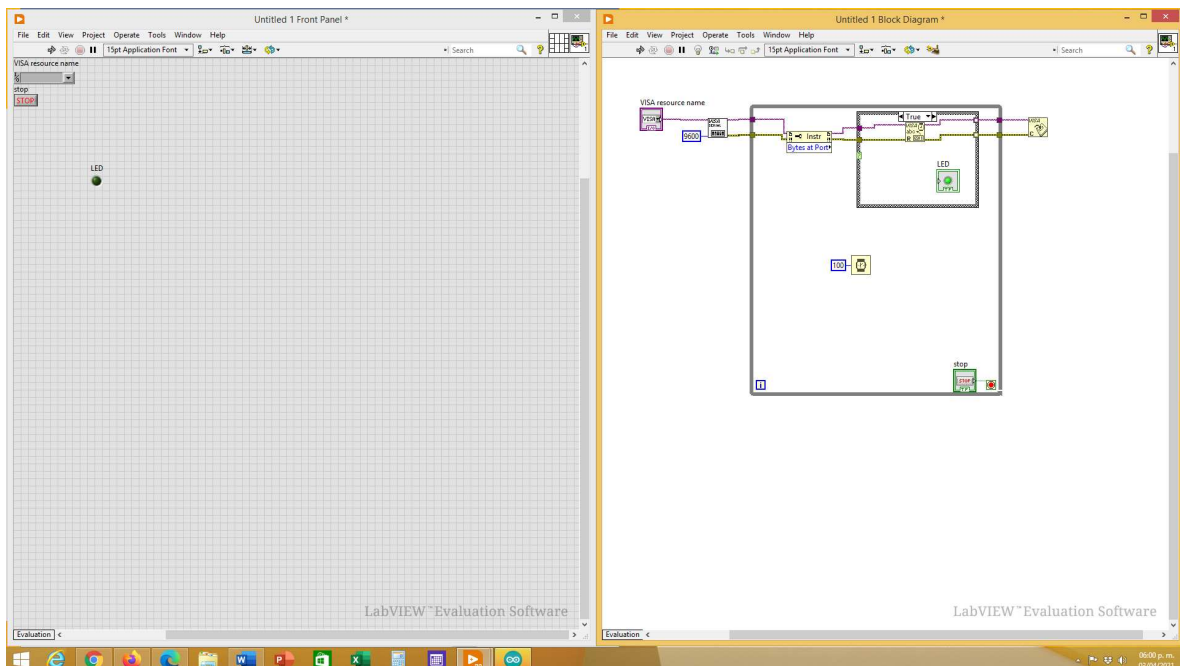
Enseguida, damos clic con boton derecho sobre ese control y elegimos crear Constante. EL valor que colocaremos ahí, sera de 100 ms



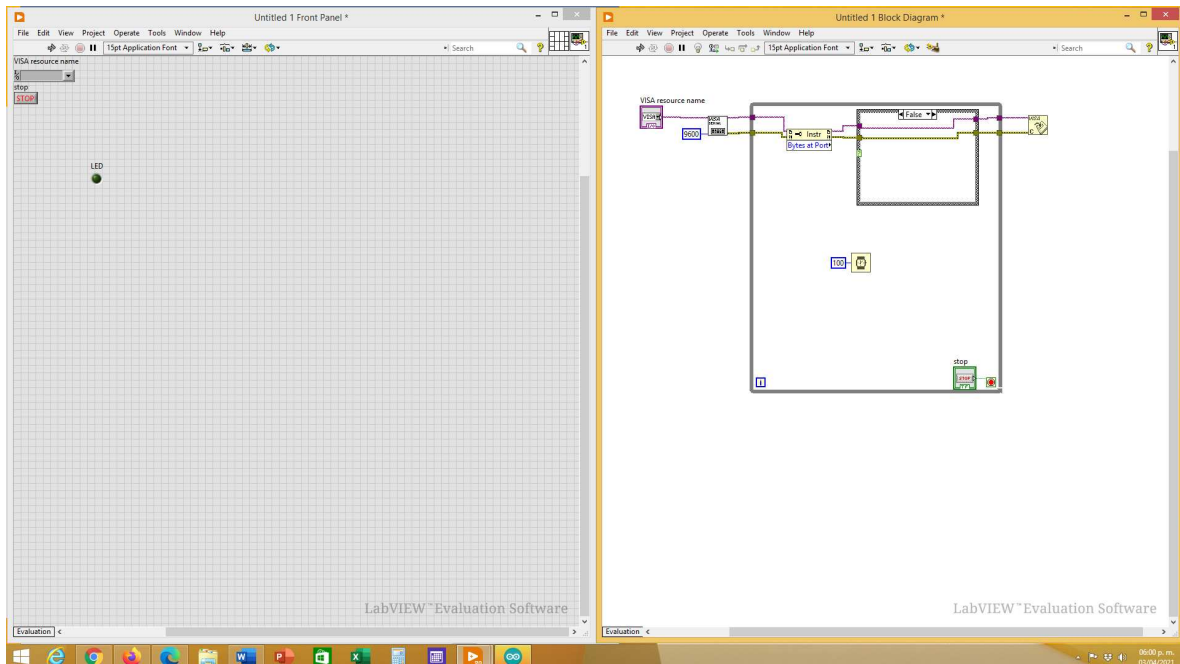
Cuando se se este monitoreando el puerto serial, debemos comprobar si existe o no un valor disponible, por lo que colocaremos una estructura CASE para verificar si hay un dato o no lo hay



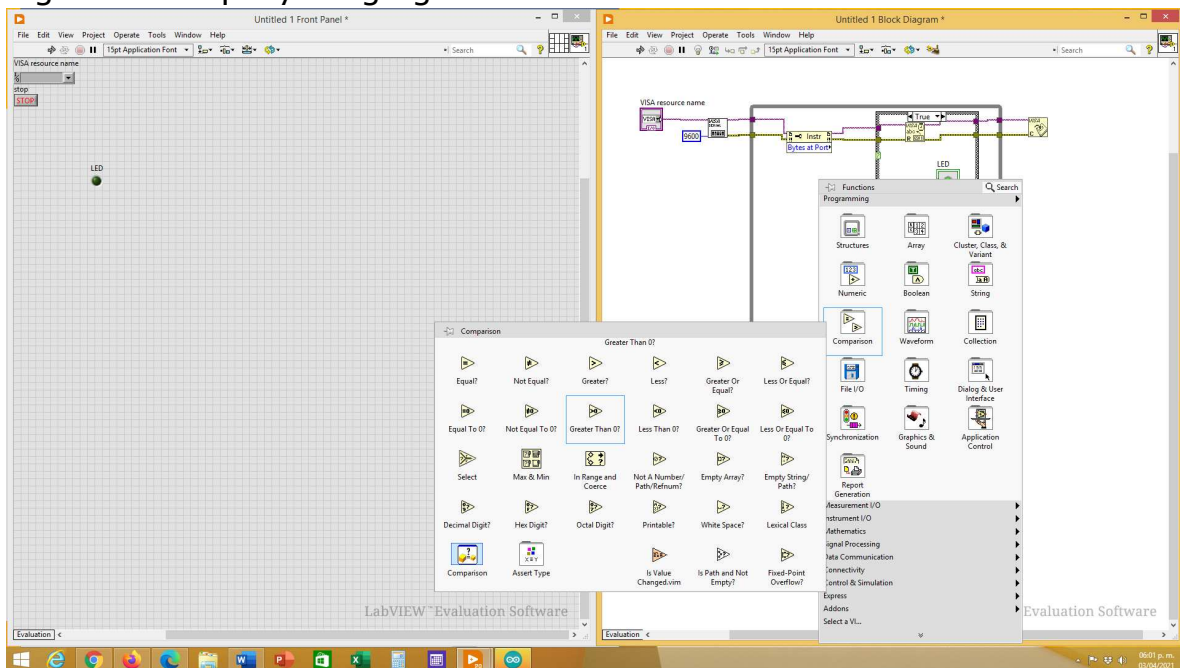
En la parte del TRUE, meteremos el bloque del LED junto con la parte del READ



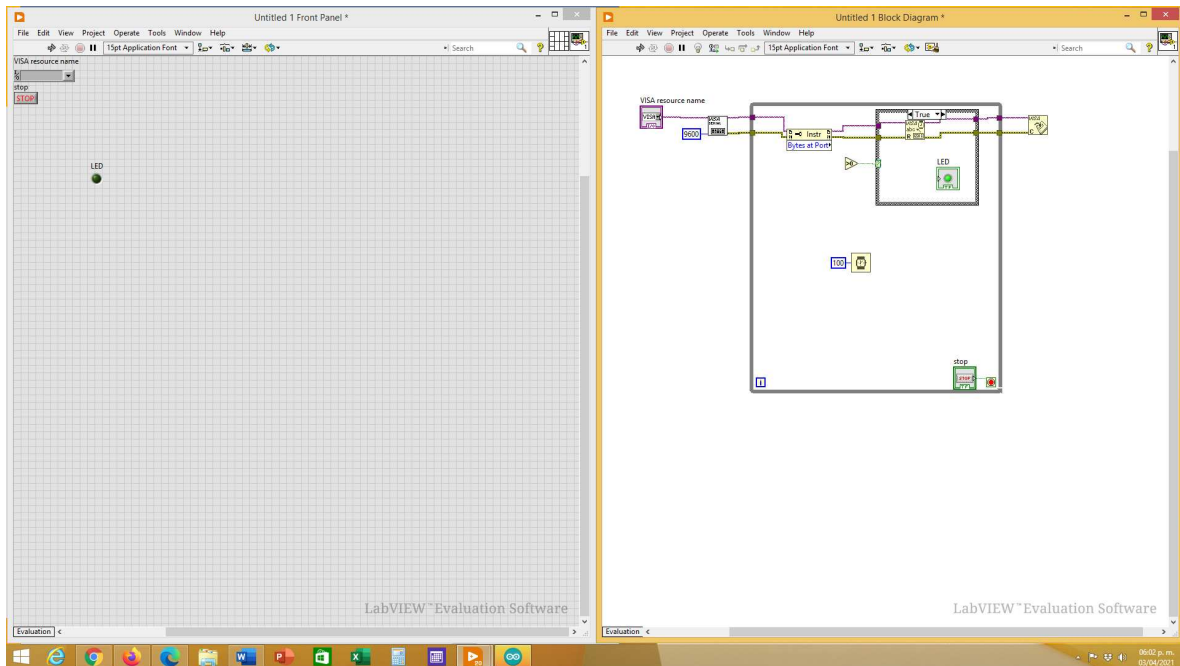
En la parte del FALSE, únicamente puentearemos las entradas con las salidas, por si no detecta datos en el puerto, que lo cierre.



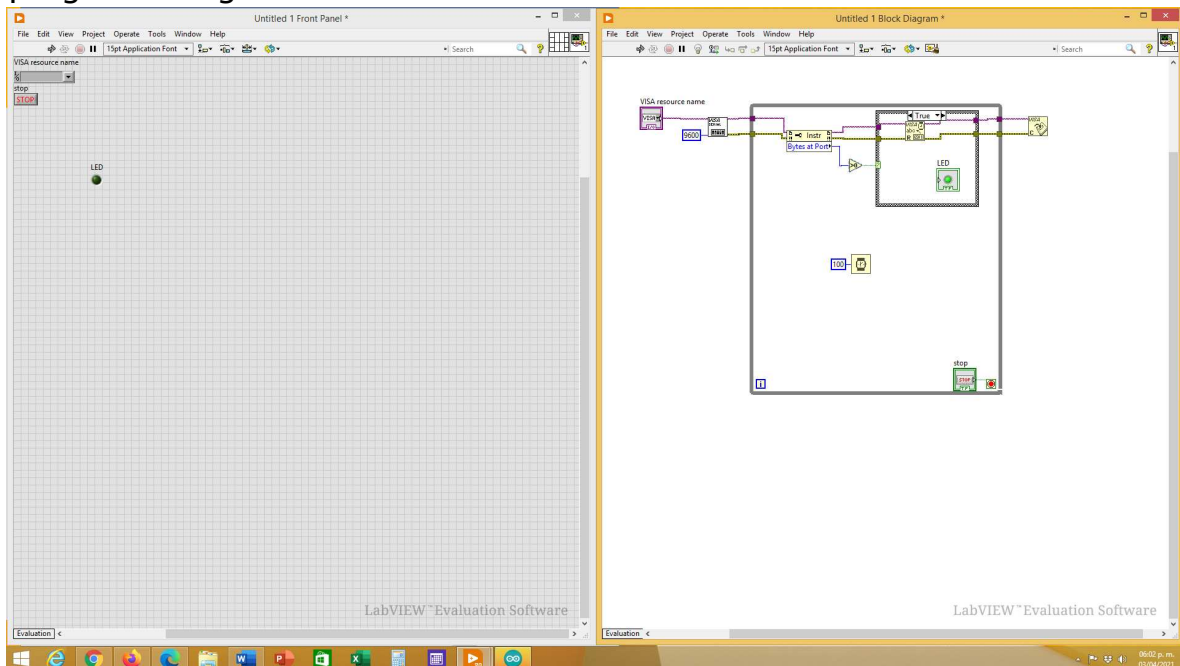
Regresando al TRUE, verificaremos si hay algun dato. Buscamos el siguiente bloque y lo agregamos



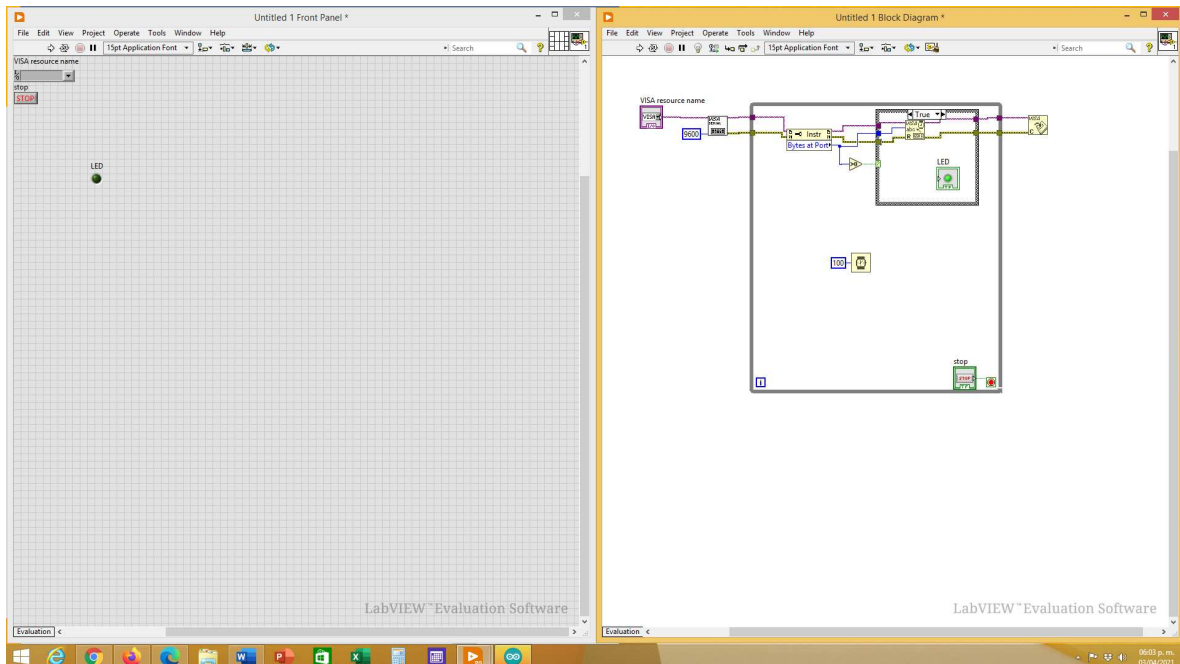
Dicha comparacion debemos conectarla en el signo de interrogacion de la estructura While como se muestra en la imagen.



Y como entrada de esa comparación, meteremos la salida del Bloque Bytes at port. Entonces si hubiera algun dato en el puerto serial entonces el programa cargaria el Caso del TRUE

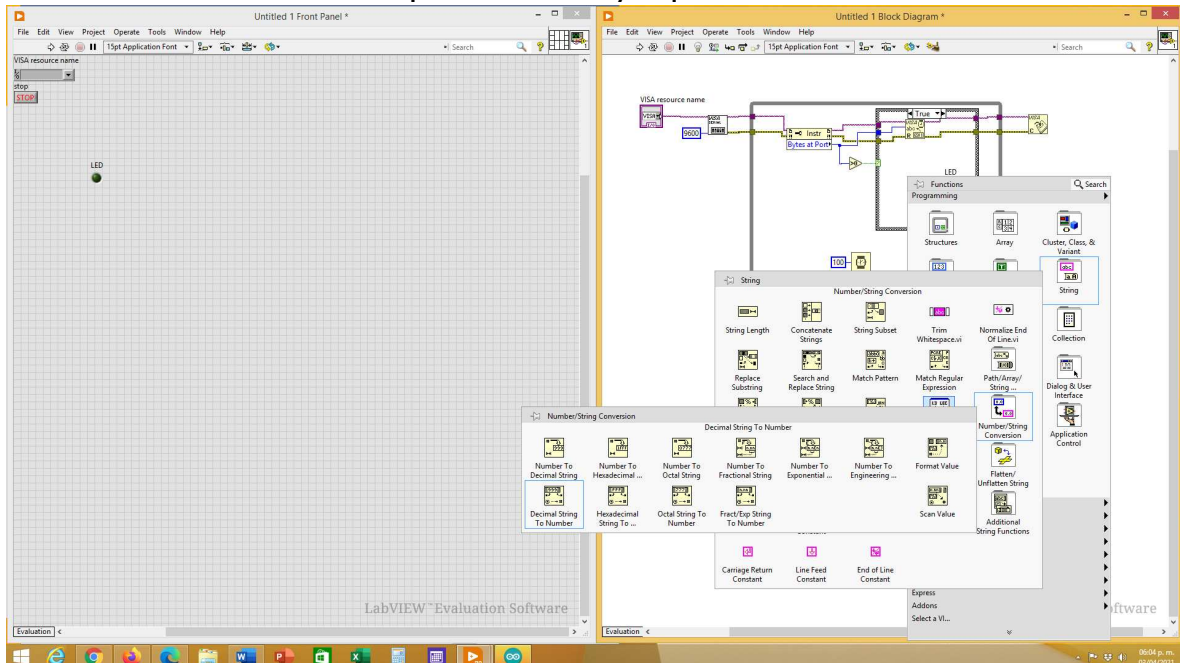


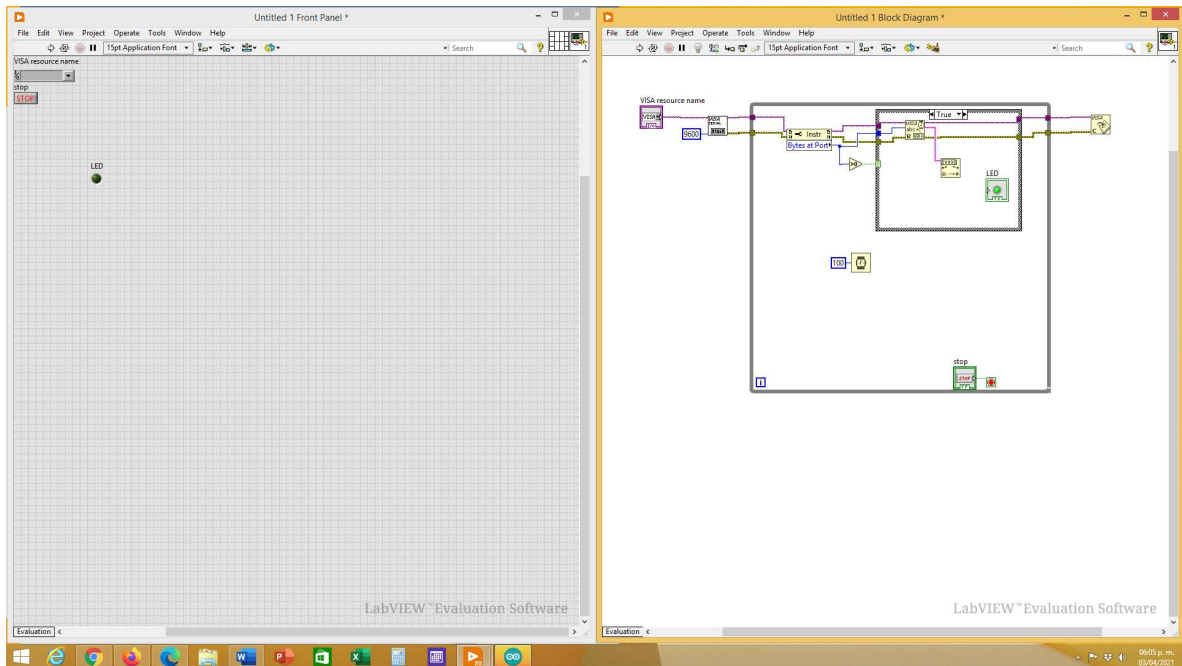
Y obviamente vamos a puentear la salida de Bytes at Port a la entrada del bloque READ



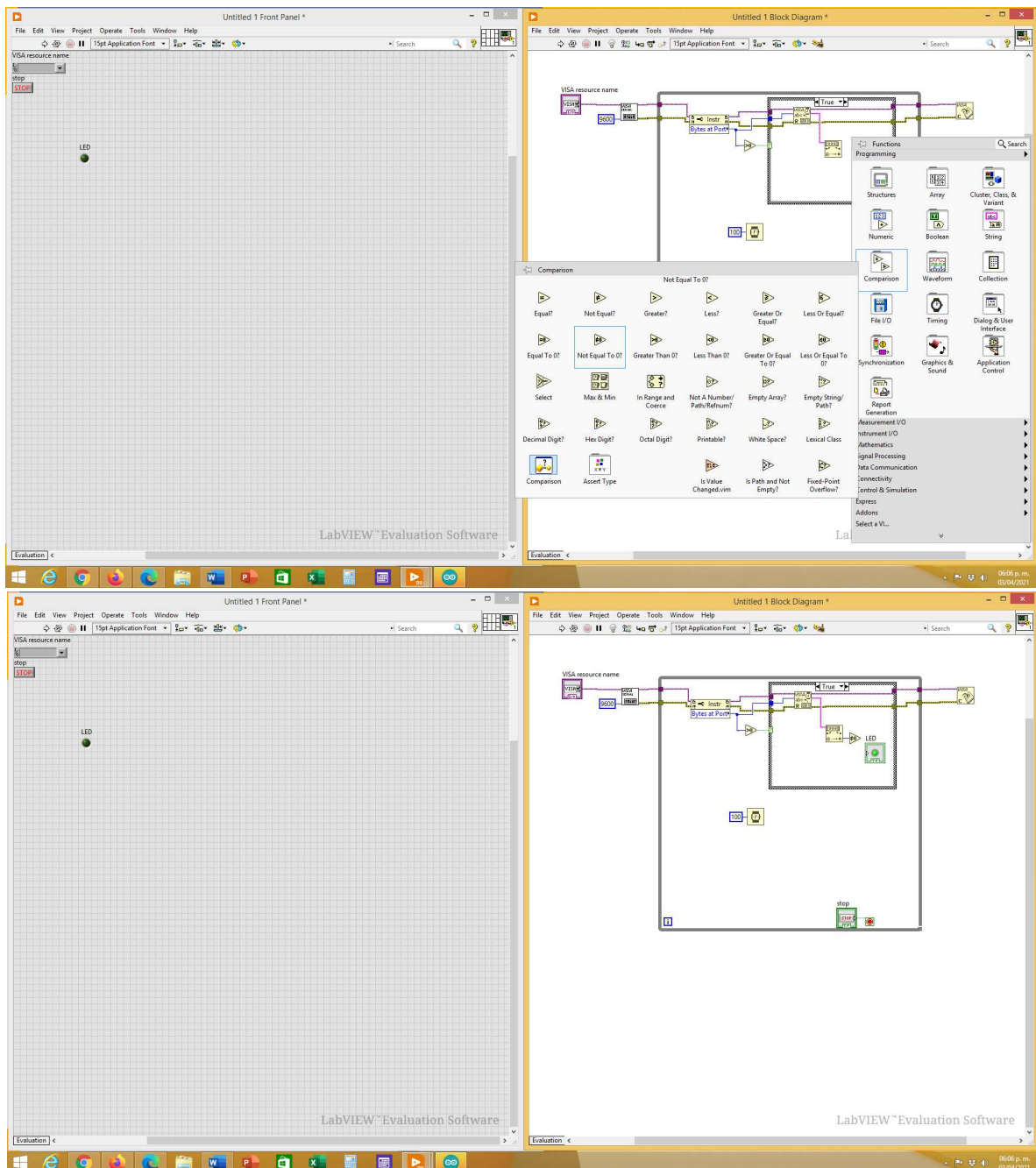
Despues conectaremos una condición para determinar si el dato que se recibe es un 1 o no lo es.

En caso de ser un 1, debera encender el led. Para esto, necesitamos convertir el dato leido a tipo numerico ya que se lee como texto

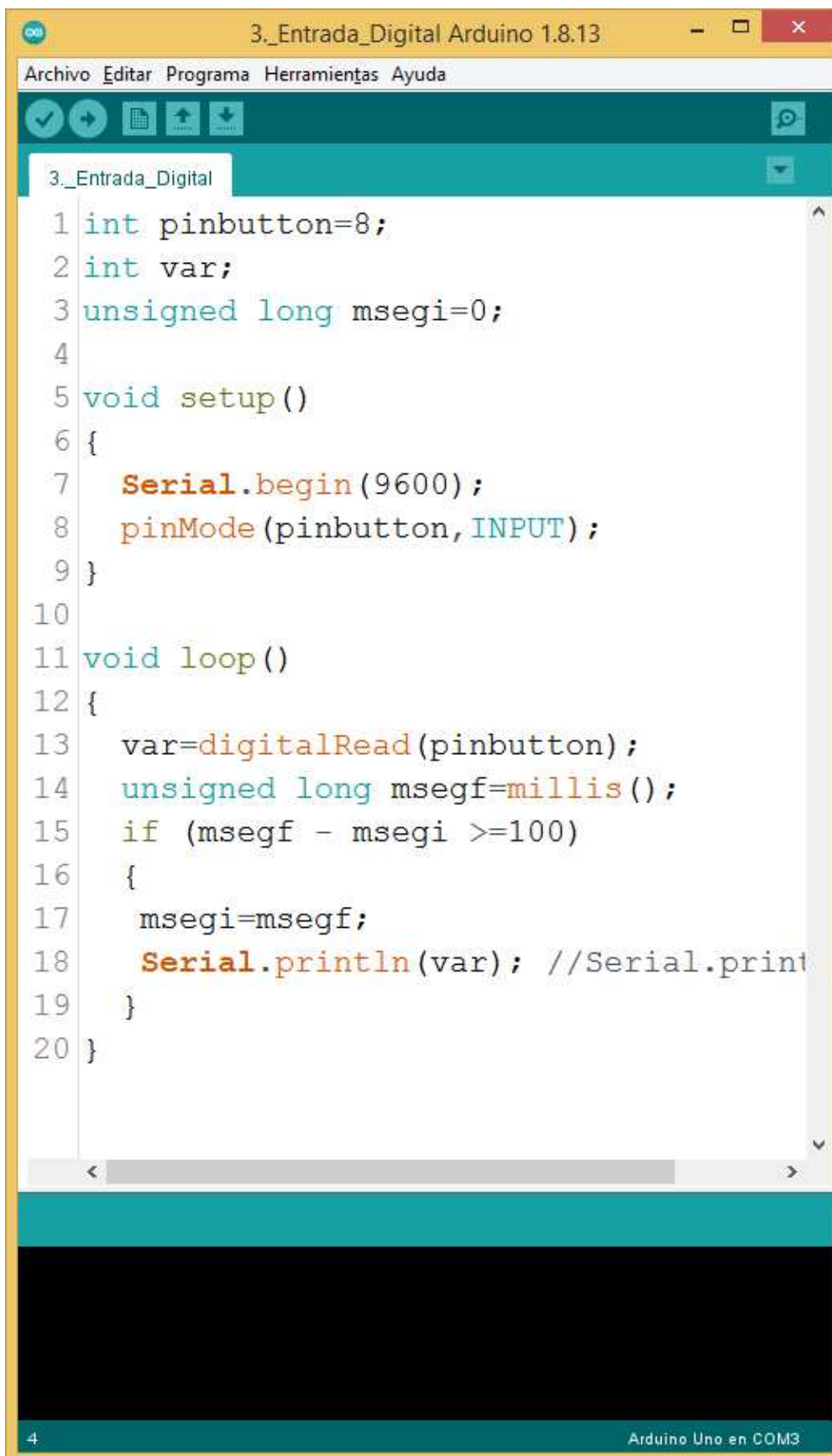




Posteriormente, verificamos si ese dato es 1 (o diferente que 0), por ello ocupamos ahora el bloque Not Equal To 0



Finalmente, la salida la conectamos al led y queda todo listo.



```
3_Entrada_Digital Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda
3_Entrada_Digital
1 int pinbutton=8;
2 int var;
3 unsigned long msegf=0;
4
5 void setup()
6 {
7   Serial.begin(9600);
8   pinMode(pinbutton, INPUT);
9 }
10
11 void loop()
12 {
13   var=digitalRead(pinbutton);
14   unsigned long msegf=millis();
15   if (msegf - msegf >=100)
16   {
17     msegf=msegf;
18     Serial.println(var); //Serial.print
19   }
20 }
```

4 Arduino Uno en COM3

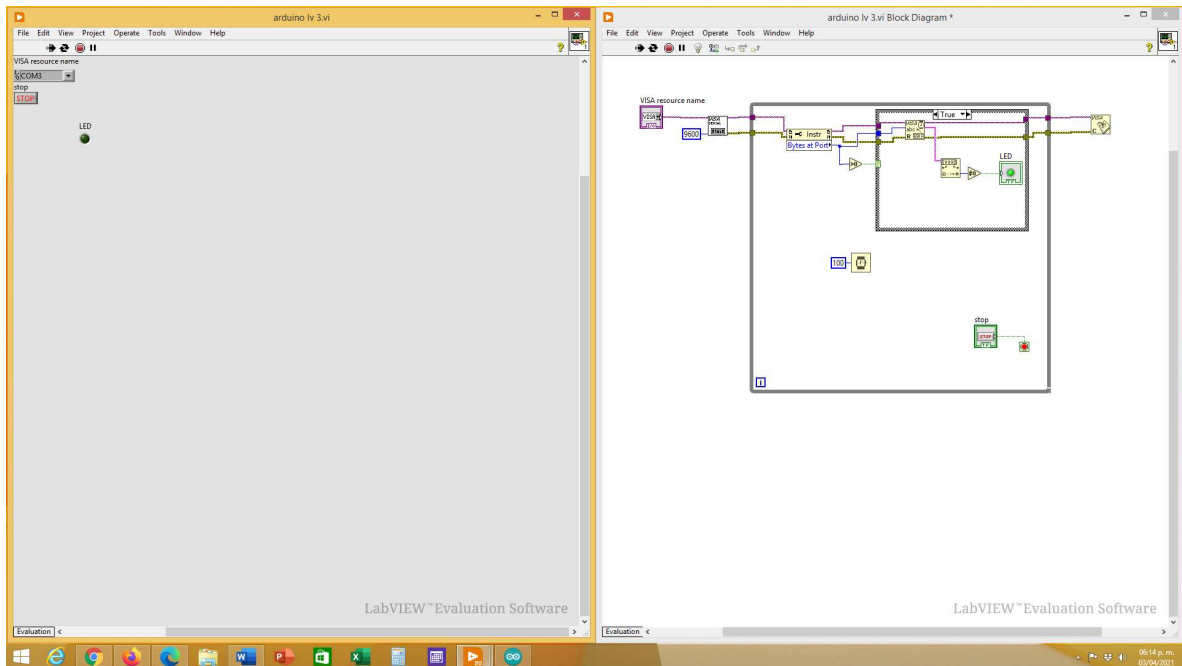
CODIGO:

```
int pinbutton=8;
int var;
unsigned long msegf=0;

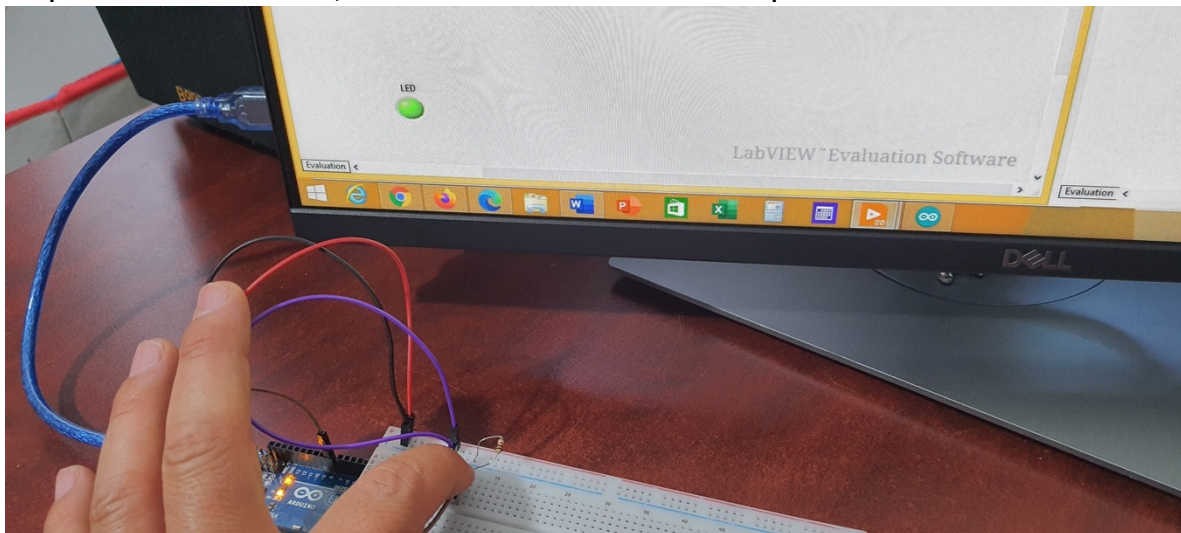
void setup()
{
  Serial.begin(9600);
  pinMode(pinbutton,INPUT);
}

void loop()
{
  var=digitalRead(pinbutton);
  unsigned long msegf=millis();
  if (msegf - msegf >=100)
  {
    msegf=msegf;
    Serial.println(var); //Serial.print(var);
  }
}
```

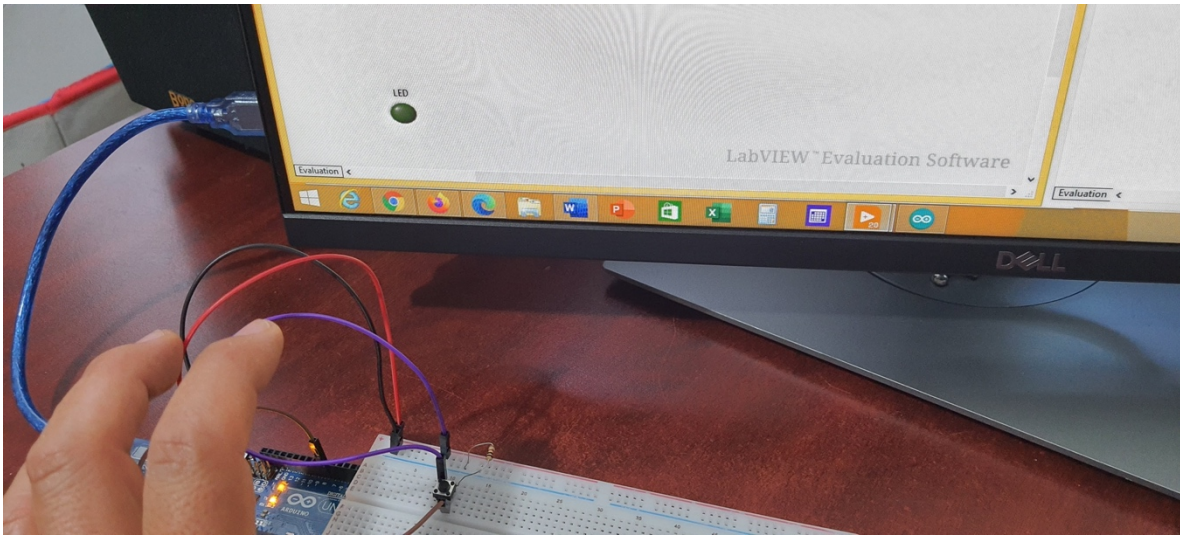
Una vez cargado el programa en la placa de Arduino UNO y conectado correctamente el interruptor, conectamos el cable USB y ejecutamos la aplicación



Al presionar le boton, veremos como el led en la pantalla se enciende.



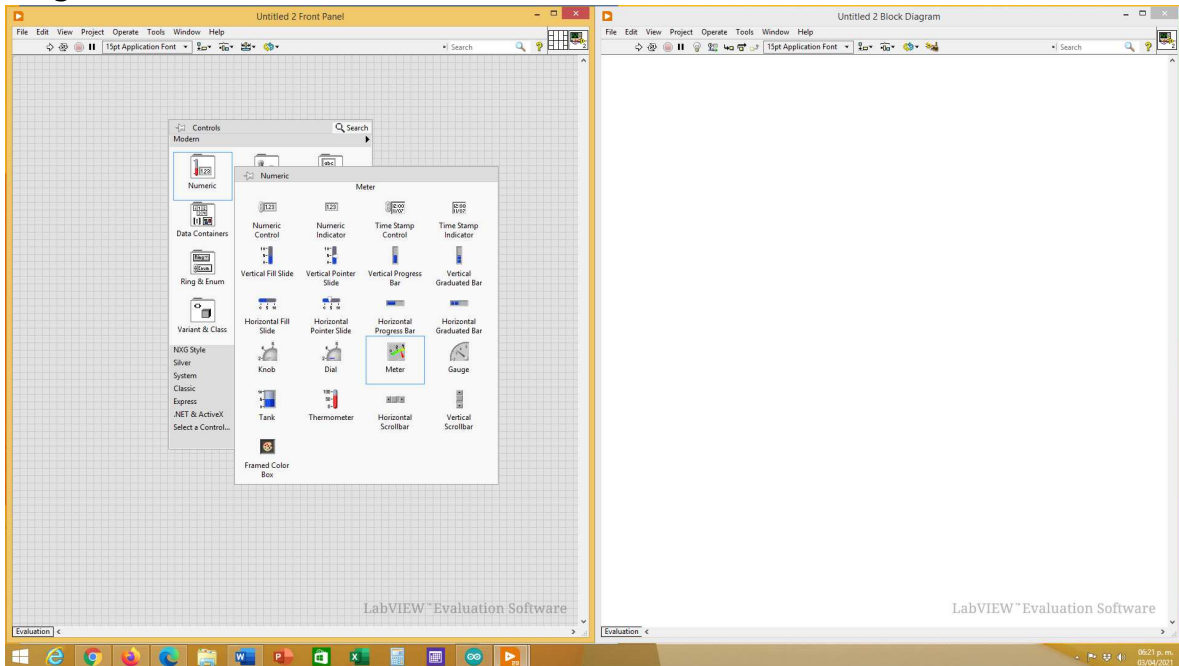
Al soltar el botón el led debera apagarse.



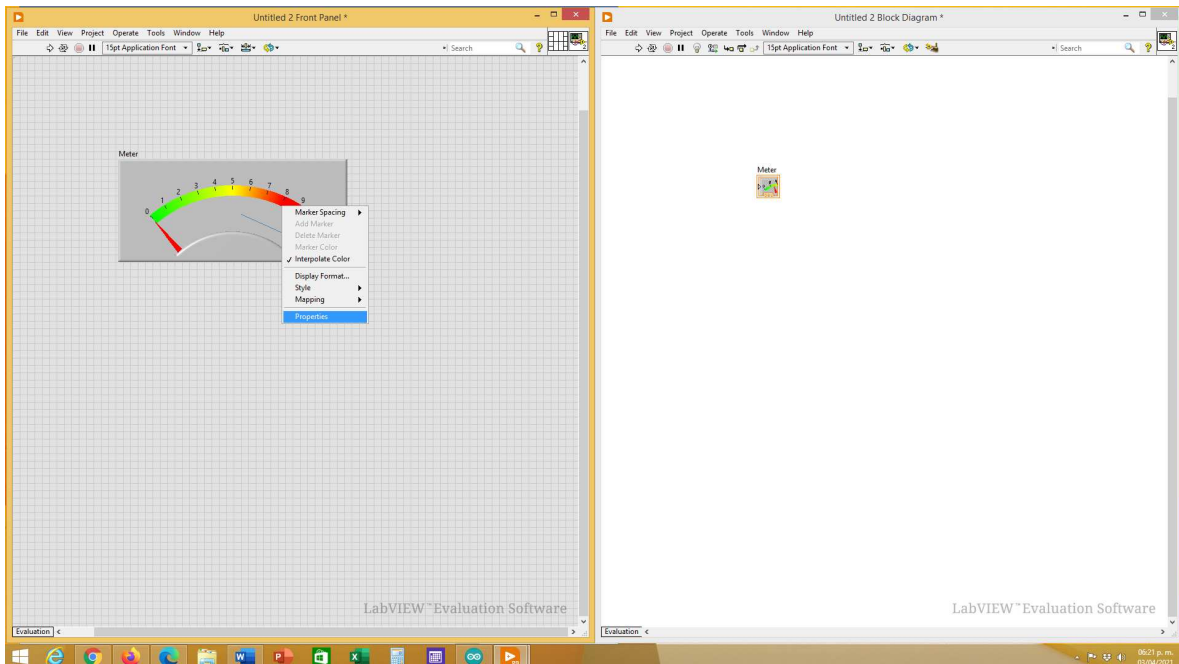
PRACTICA 13: Leer un dato analogico desde Arduino y mostrar en un indicador tipo aguja en LabView

Necesitamos abrir un nuevo Proyecto tipo VI

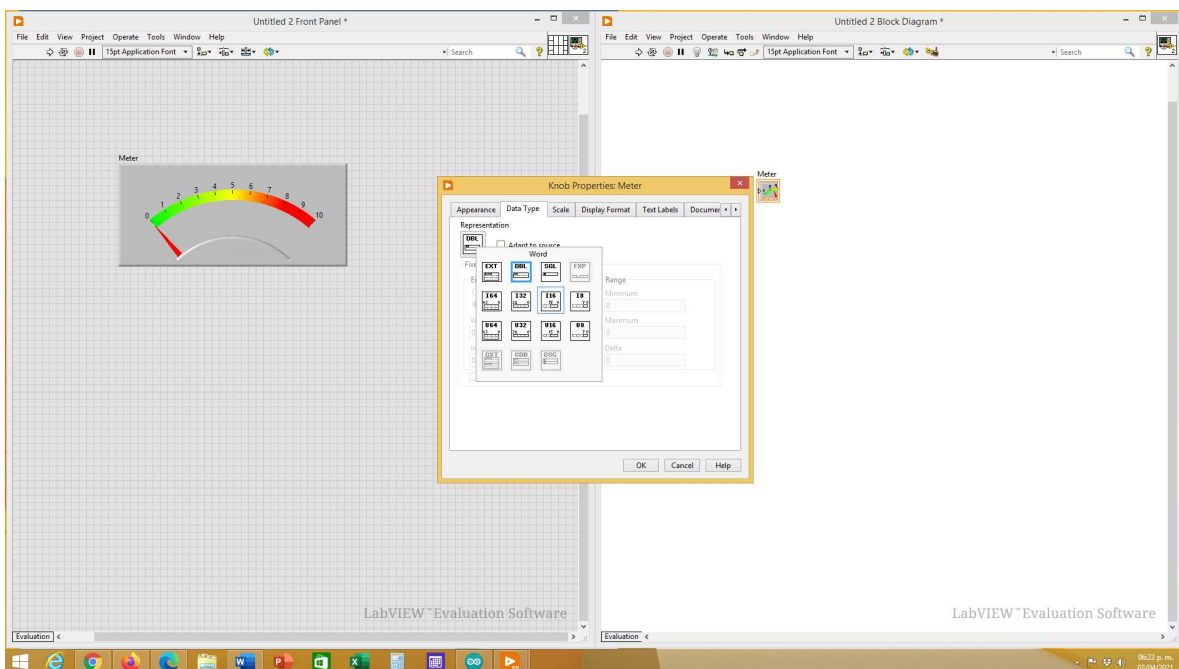
En el panel Frontal, insertaremos un Control de Monitoreo Numerico, elegimos entonces Meter



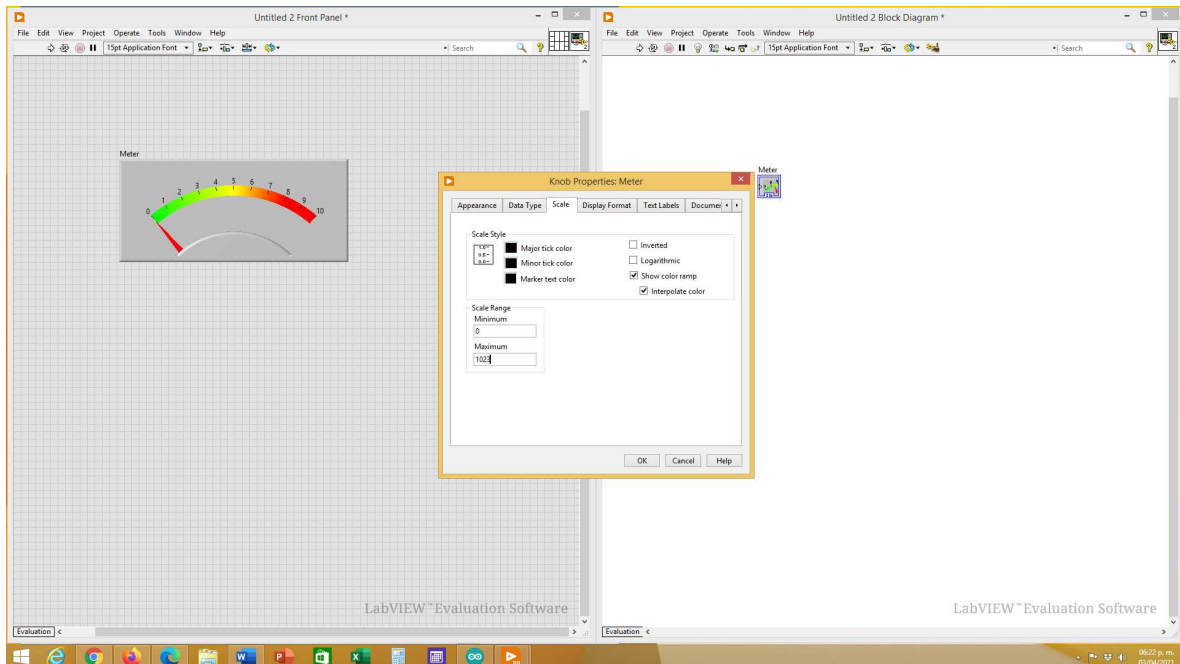
Una vez que este agregado, veremos que en la ventana de Bloques, aparecera el bloque correspondiente.



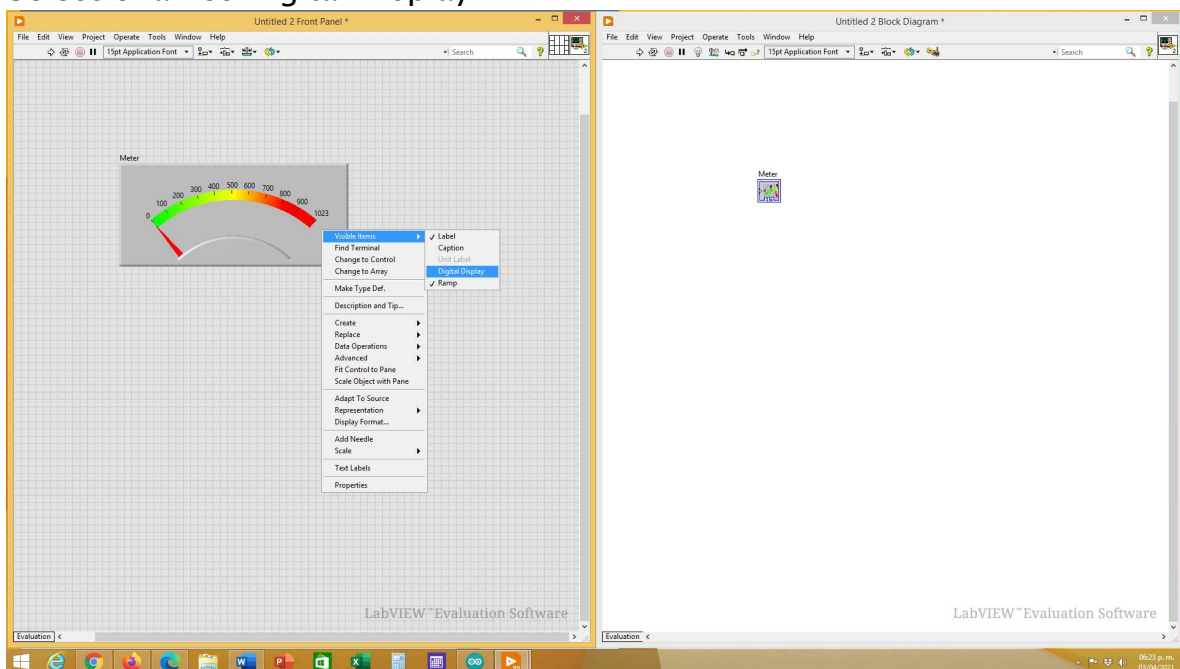
Damos clic secundario sobre el control Meter y elegimos Propiedades. En la ventana que se abra, seleccionamos la pestaña Data Type y Elegimos DBL



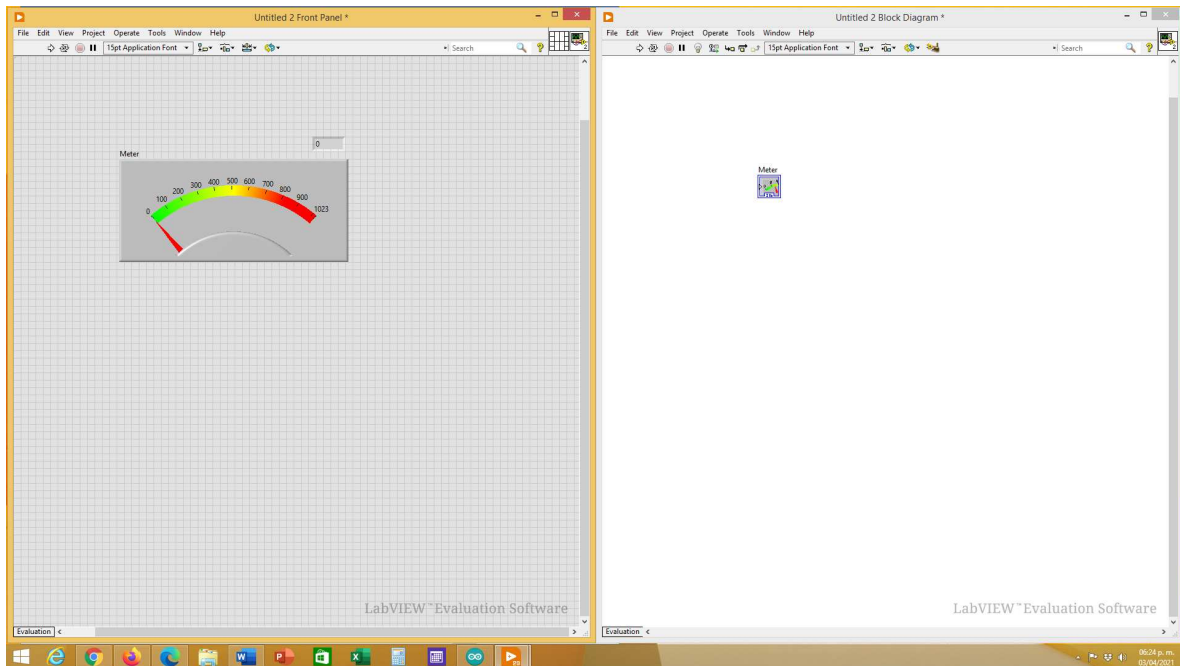
En Scale, vamos a darle el valor minimo de 0 y el maximo de 1023 al Rango de valores. Estos valores debido a que Arduino va a leer valores analogicos desde el potenciometro y deben corresponder con esos datos.



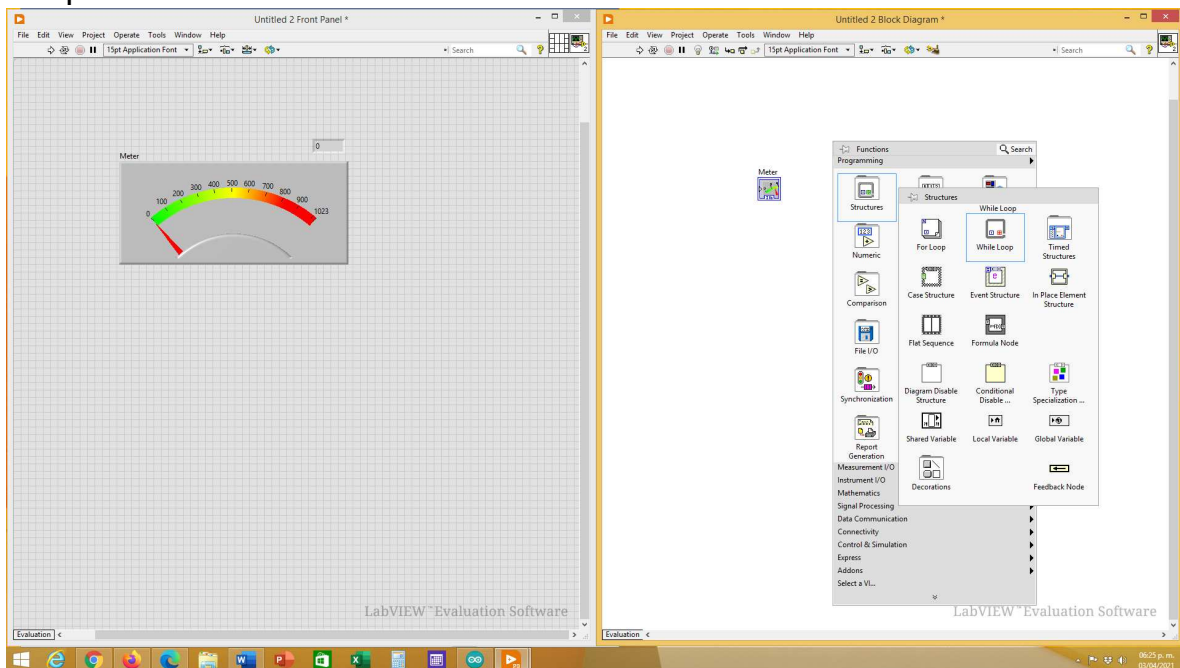
Posteriormente, damos clic en el boton OK. Despues nuevamente clic secundario sobre el control Meter y aparecera la opcion Visible Items, Seleccionamos Digital Display



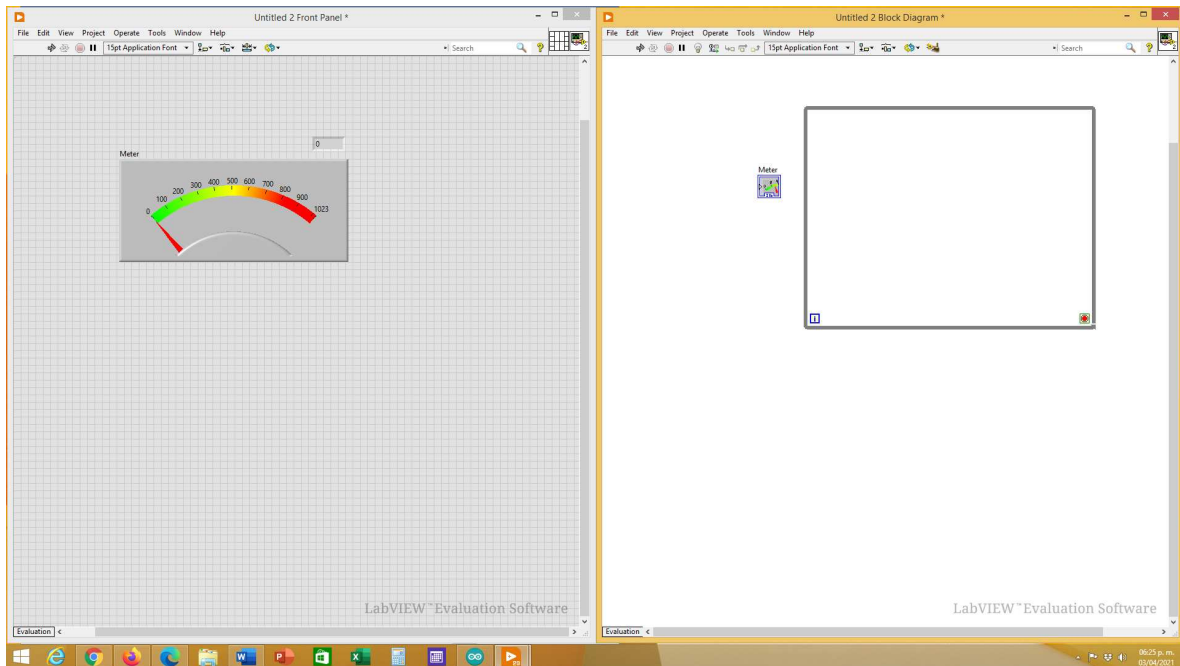
Aparecera un pequeño Display con un valor. Aquí veremos los valores que vayamos enviando desde arduino al mover el potenciómetro.



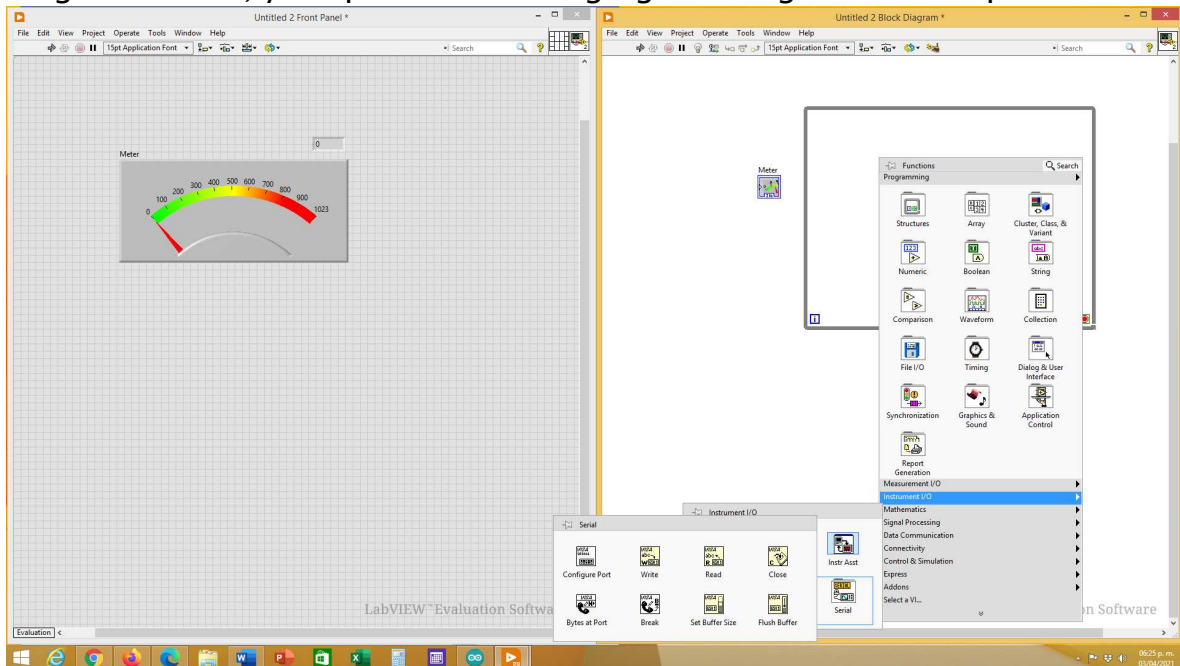
En la ventana de los Bloques, vamos a Agregar una Estructura tipo While Loop



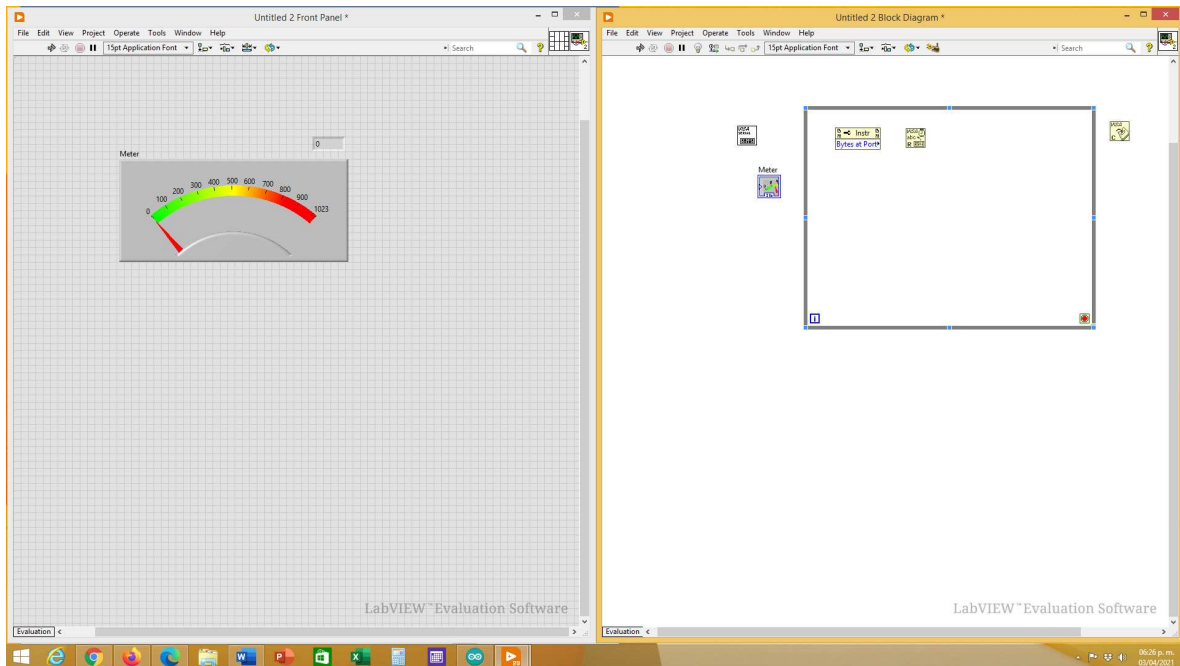
Una vez agregara, dentro vamos a meter los controles para comunicarnos con el puerto Serial.



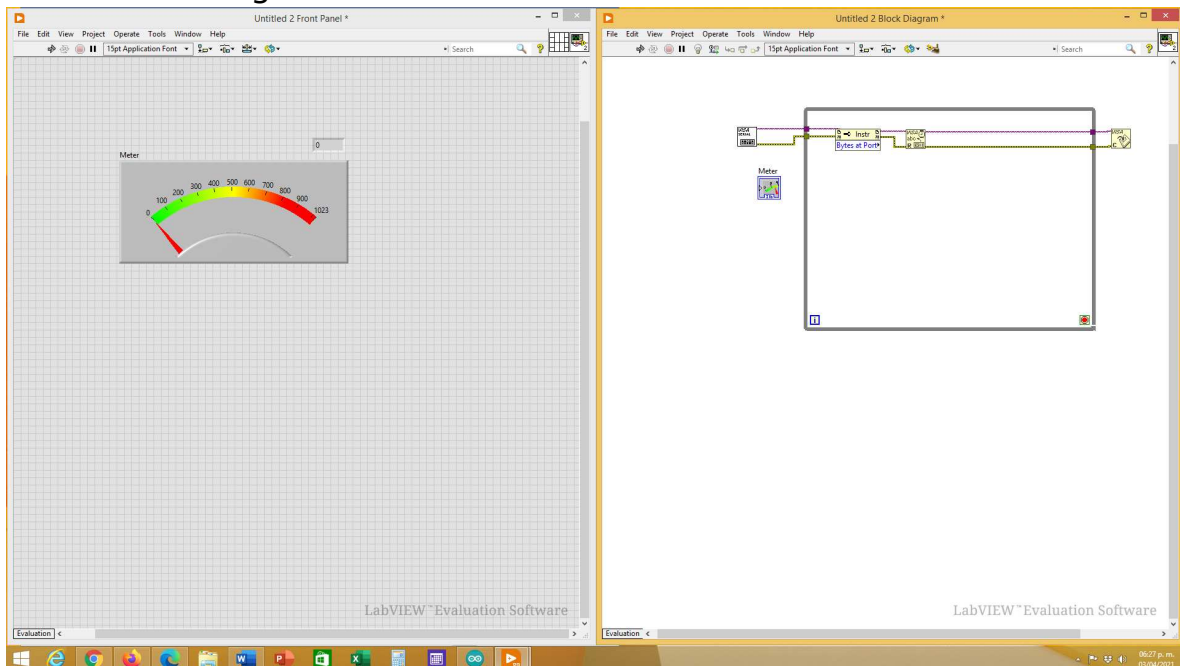
Para esto damos clic con boton derecho y en el menu Instrument I/O, elegimos Serial, y despues vamos a agregar los siguientes componentes



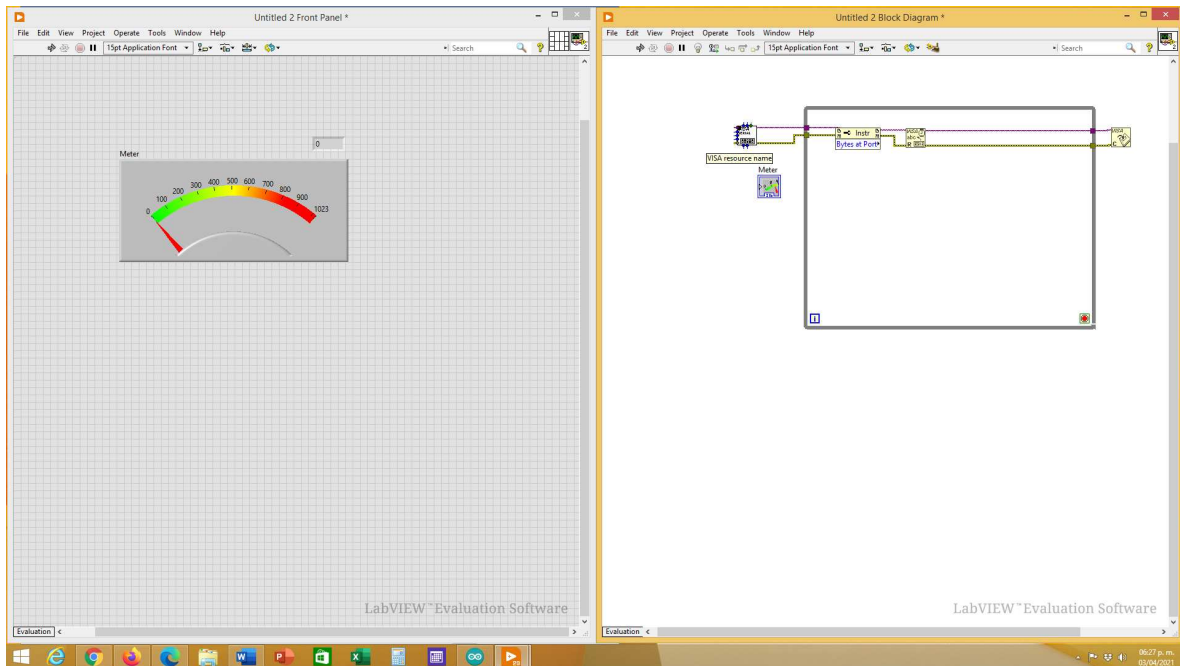
VISA Serial, Bytes at Port, Read y Close



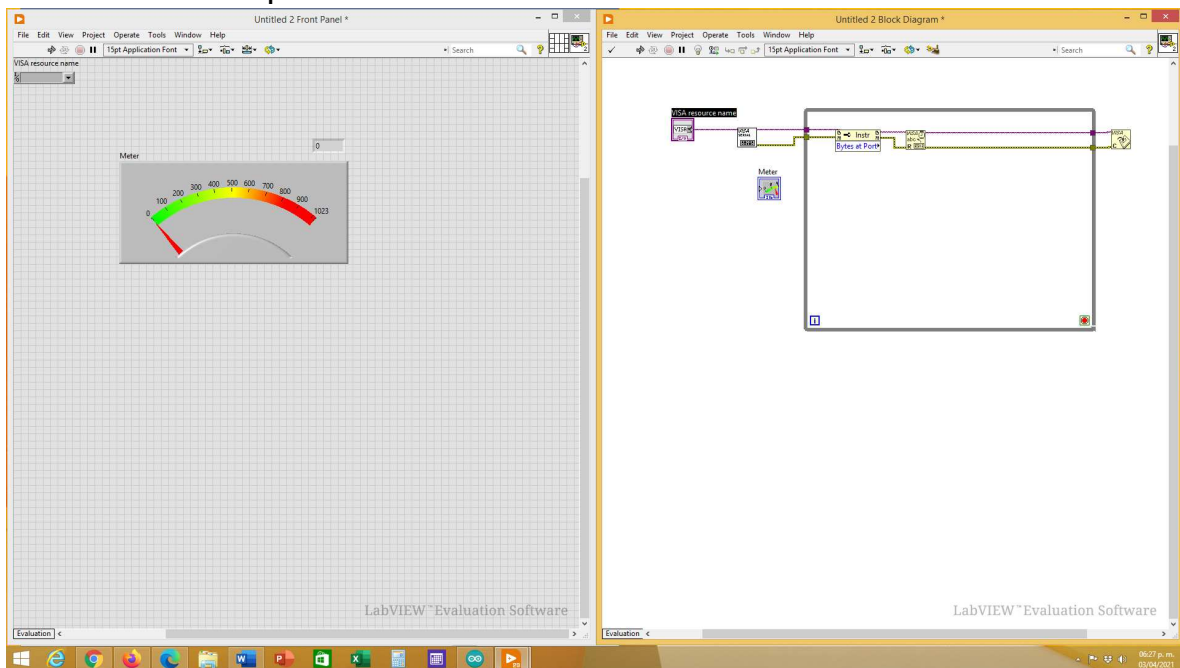
Conectamos las entradas y salidas de cada bloque, observando que se crean tuneles con el marco del While Loop. Conectaremos como se muestra en la figura.



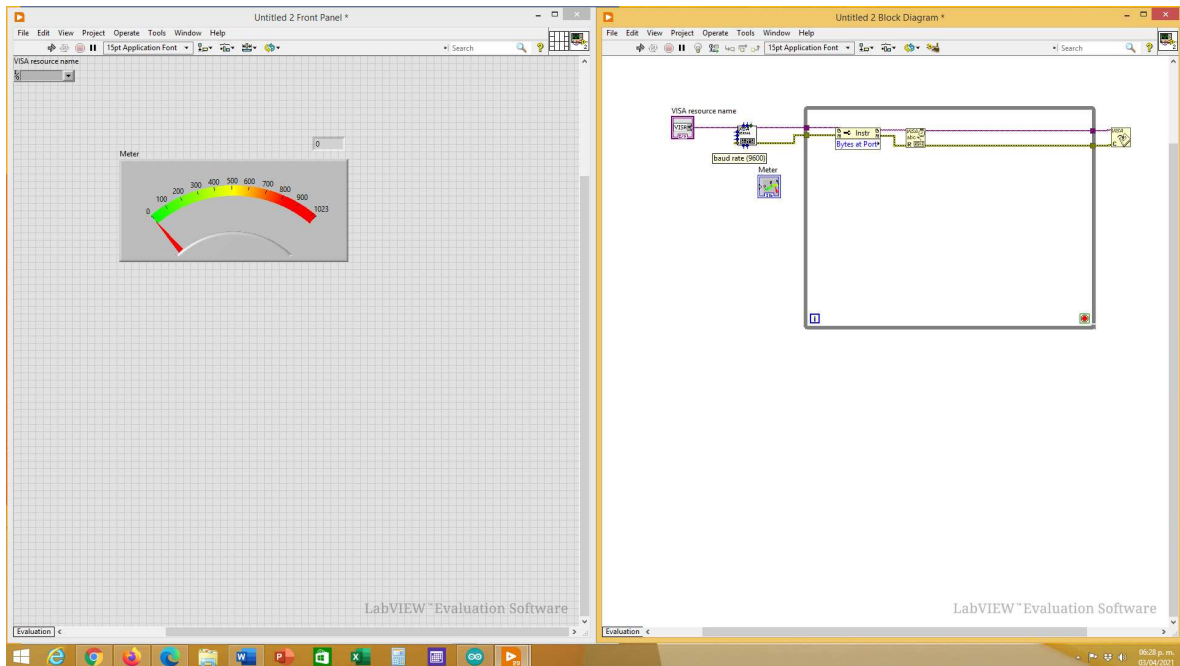
En el control VISA SERIAL, agregaremos el control para seleccionar el puerto que debe corresponder al mismo que usa Arduino. Para esto damos clic secundario en el primer conector de arriba y Agregamos el Control



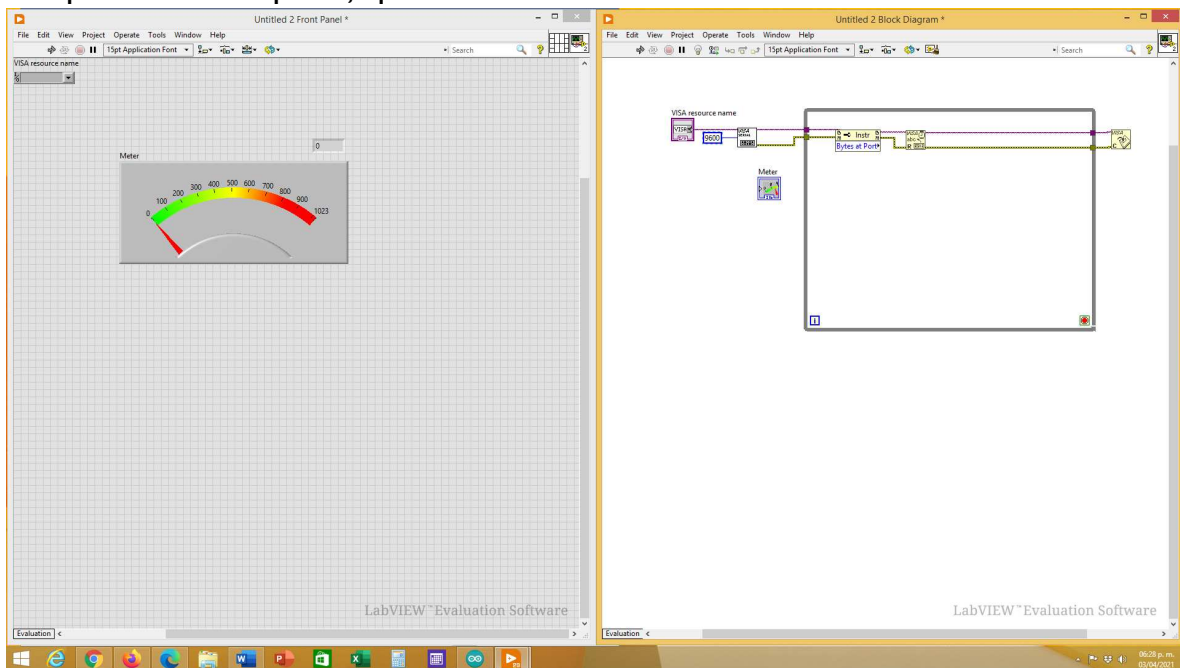
Veremos como aparece en el Panel Frontal dicho control.



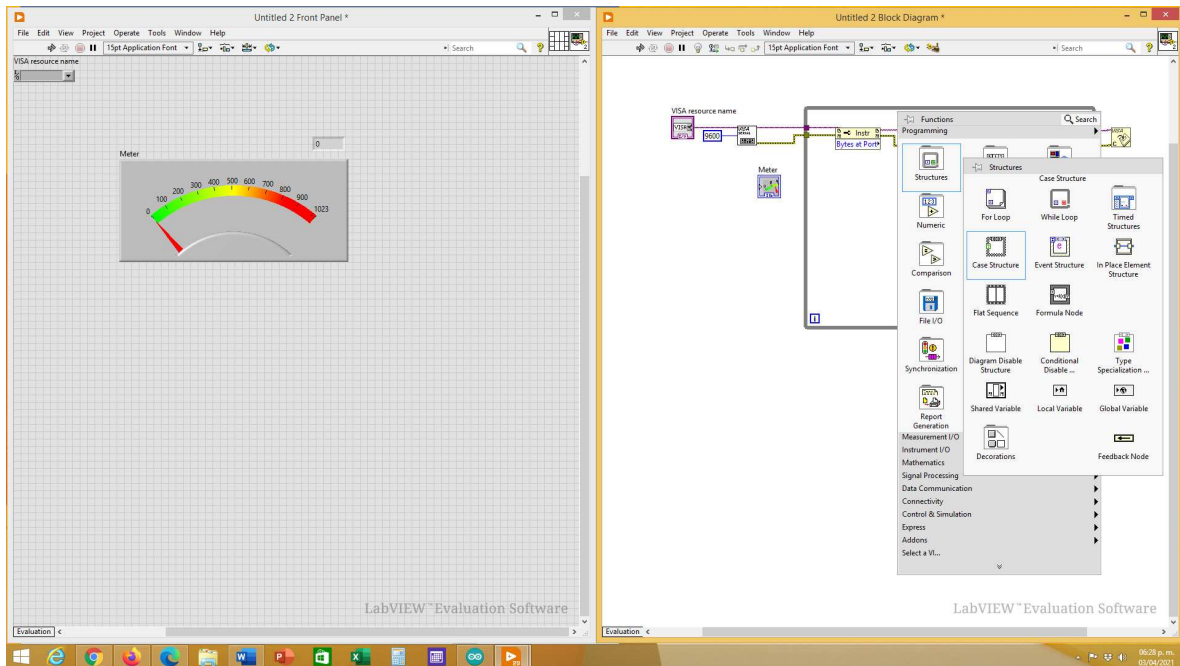
Despues vamos a dar clic sobre el siguiente conector de VISA SERIAL y vamos a elegir Constante para darle un valor de 9600 que sera la velocidad de comunicaci3n en baudios que sera la misma que tenga el puerto serial del arduino. Entonces colocarmos el valor de 9600



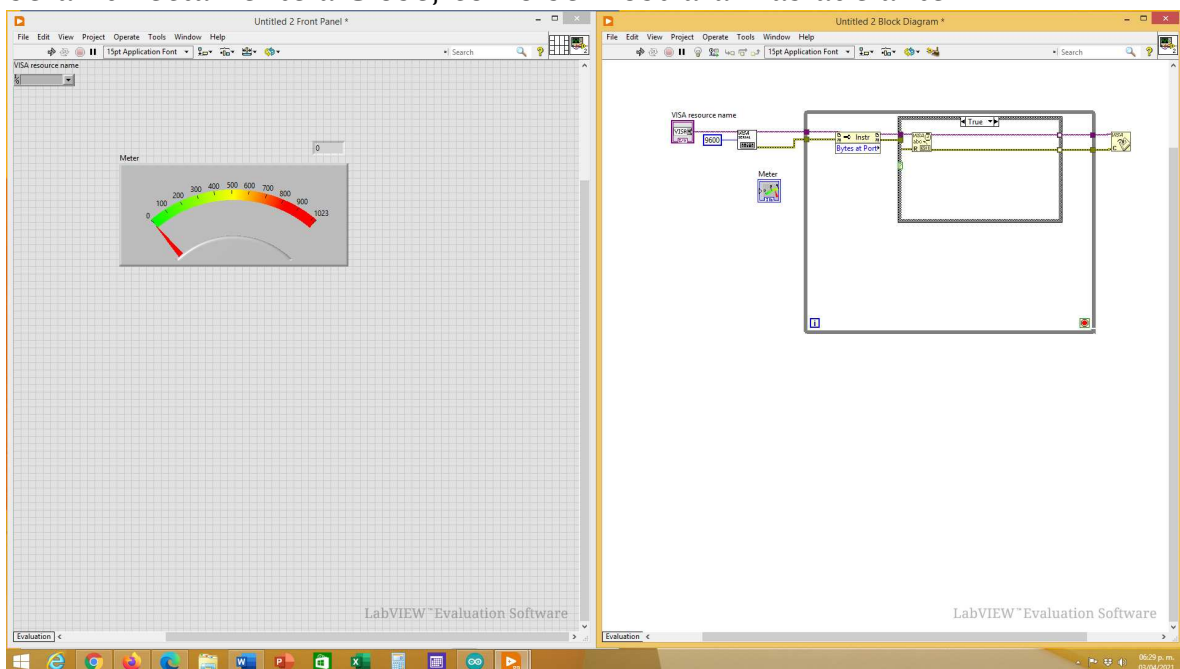
Despues de este paso, quedara asi:



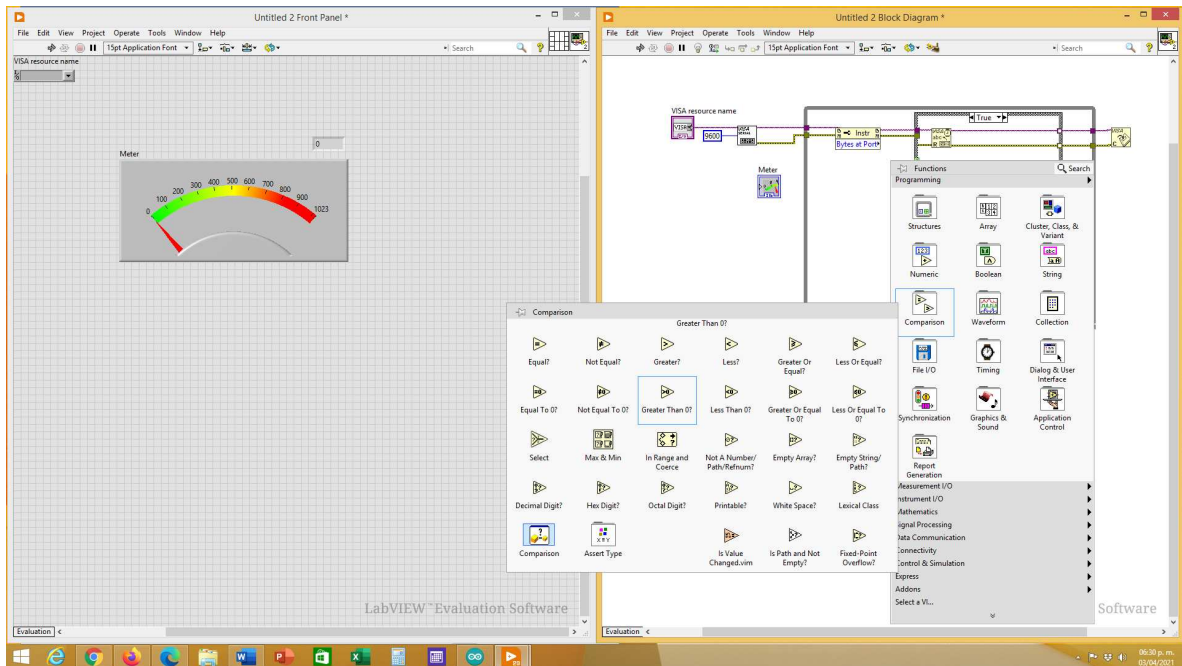
Ahora vamos a meter un CASE para que esta lectura se realice solo cuando haya un dato en el puerto serial, en ese caso el dato sera leido y enviado al control Meter del panel Frontal



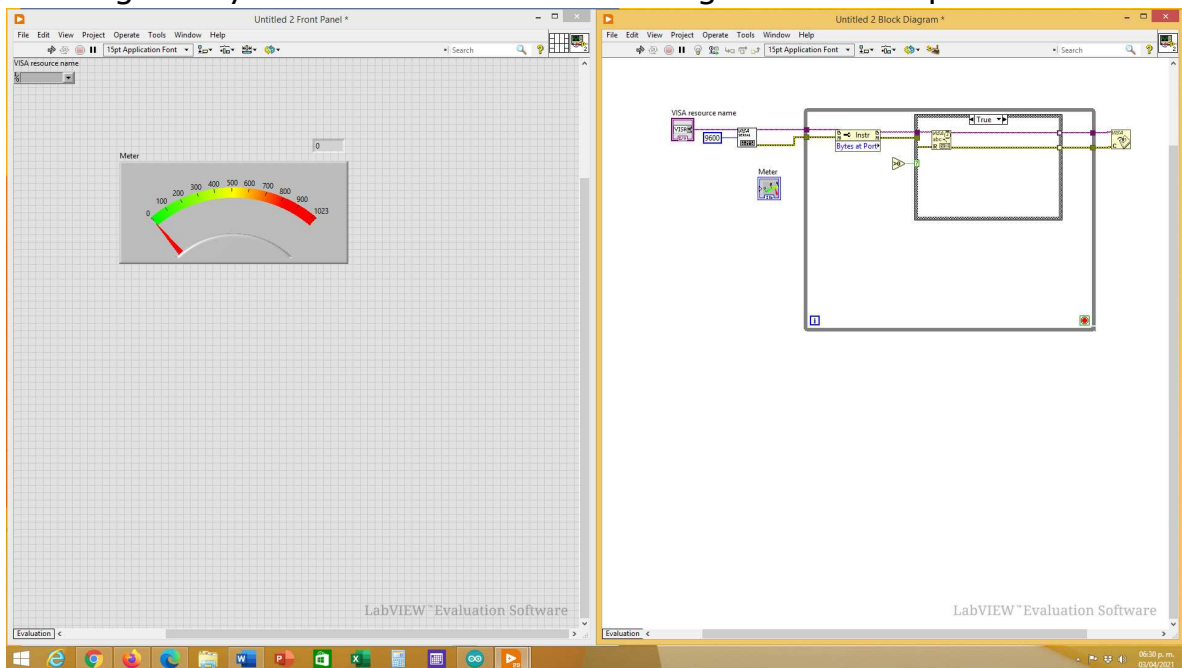
En la opción de TRUE, realizaremos las conexiones normales ente Bytes at Port, READ y Close. En el Caso de FALSE, no habra lectura y las conexiones seran directamente a Close, como se mostrara mas adelante



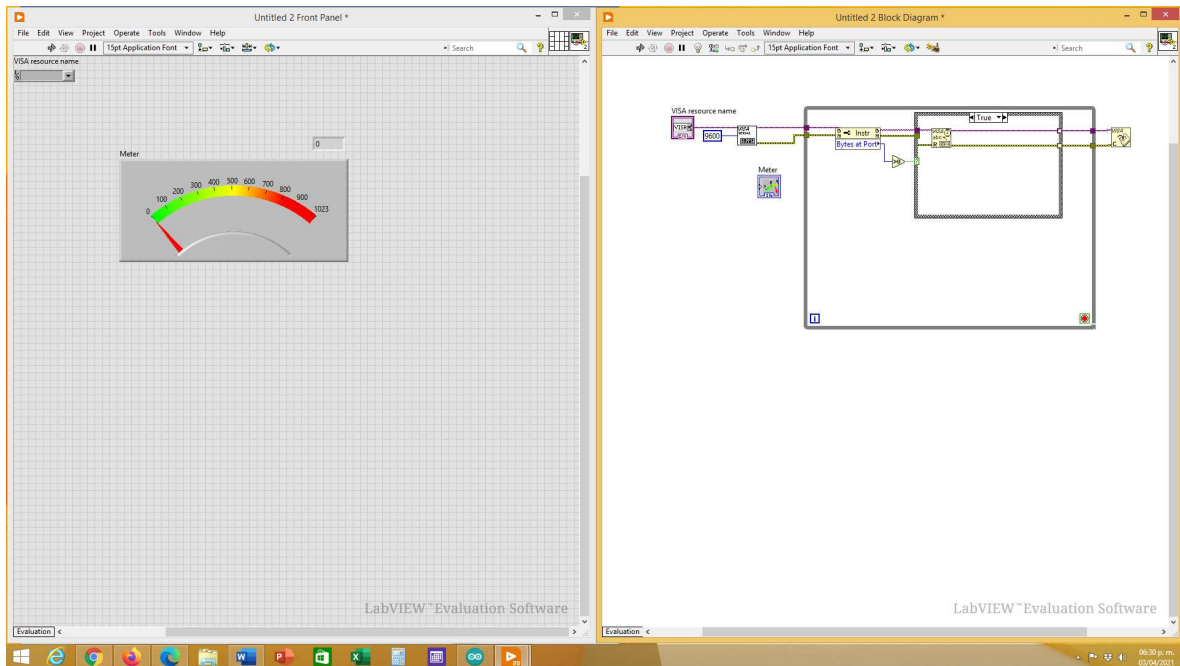
Agregaremos un bloque que determinara si hay un valor en el puerto serial,
este bloque correspondera a la condicion si es mayor a 0:



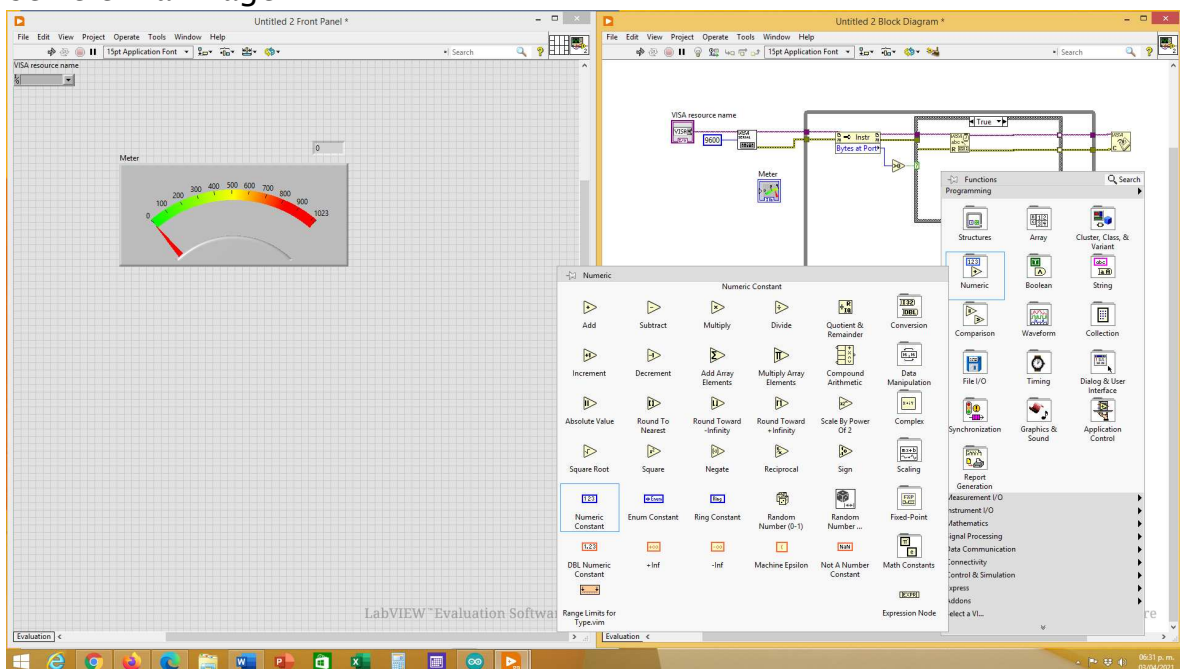
Lo afegamos y lo conectamos con la interrogación del bloque CASE



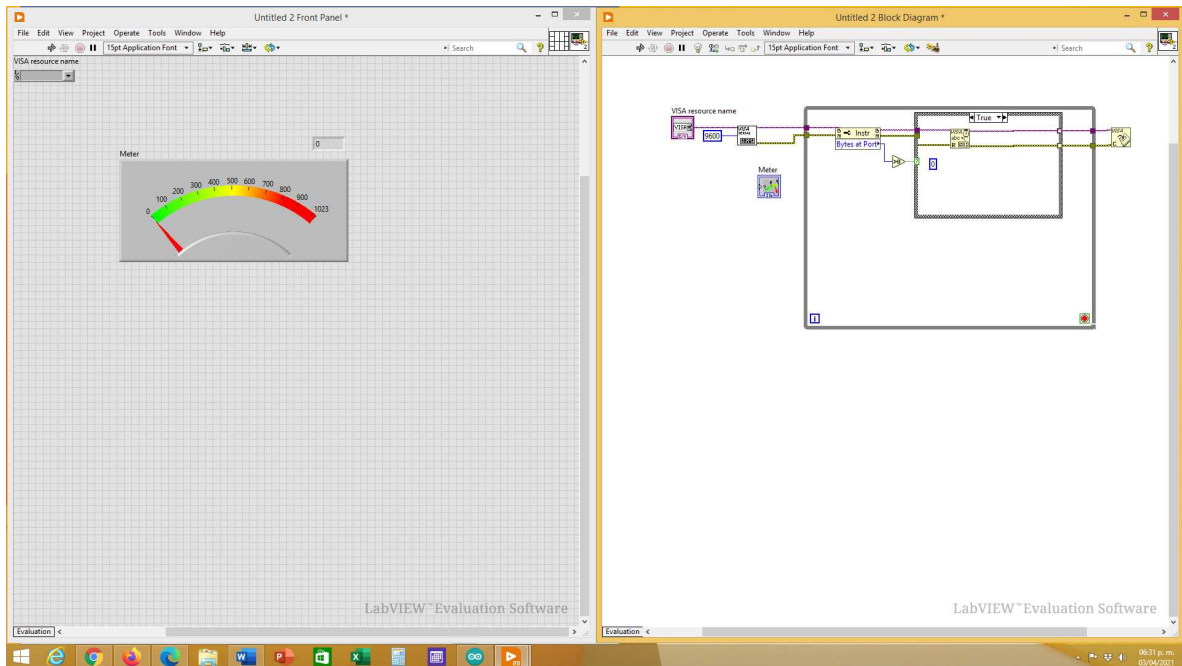
La entrada de esta condición vendra de la salida de Bytes at Port



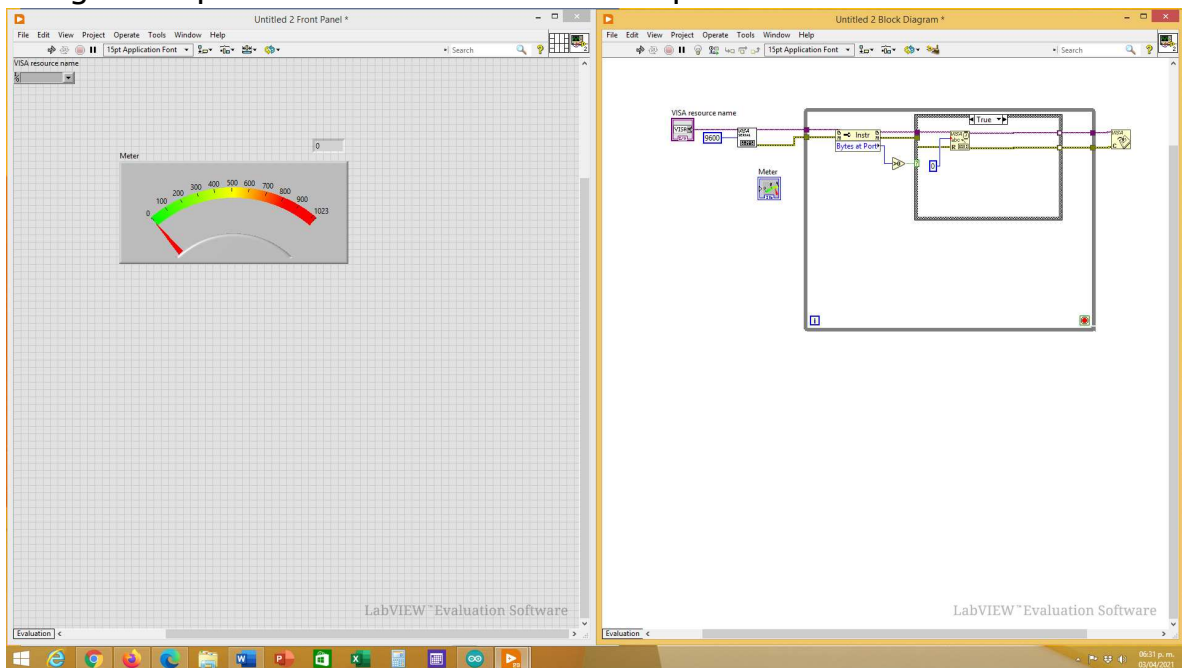
El siguiente paso será agregar un numero que nos servira para comparar posteriormente, para esto agregaremos el control Numeric Constant como se ve en la imagen



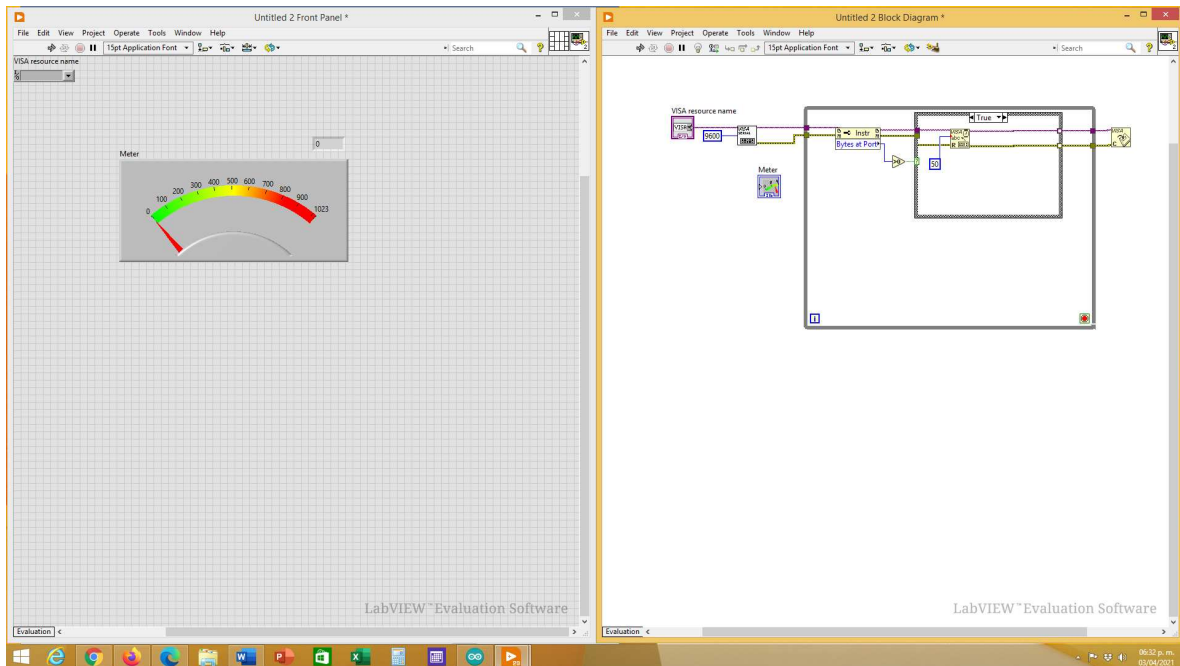
Este numero que sera un 50, lo metemos en el CASE (inicialmente al agregarlo, por default aparece como un 0)



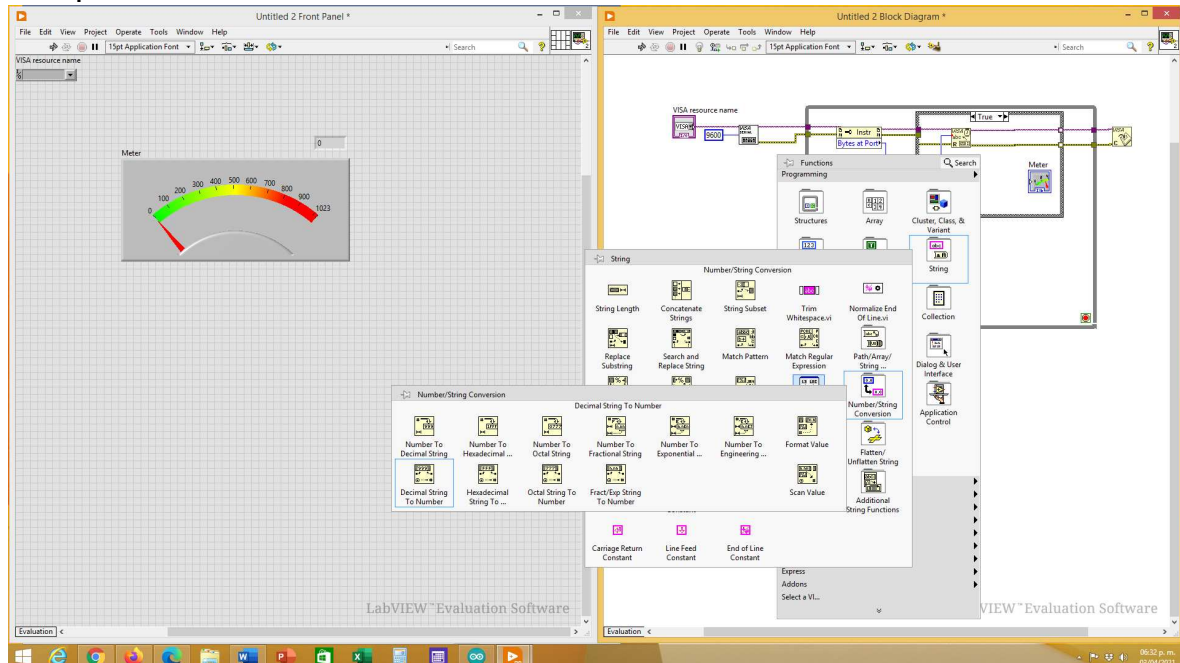
El siguiente paso sera introducirlo a el bloque READ



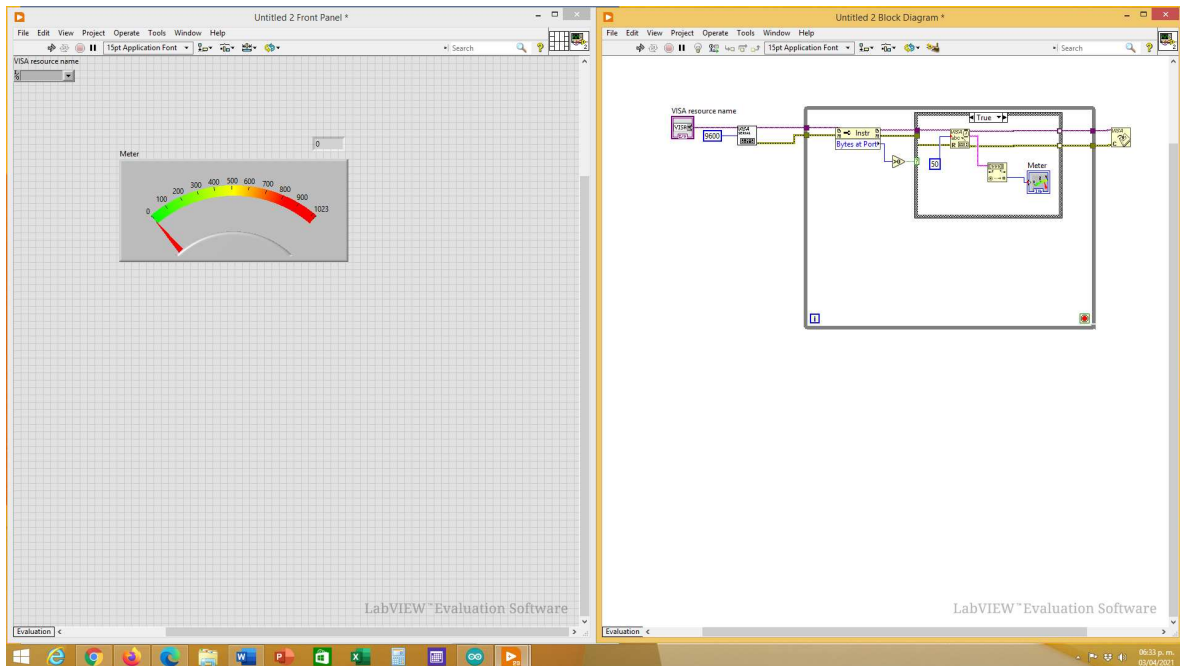
Despues le cambiamos el numero a 50



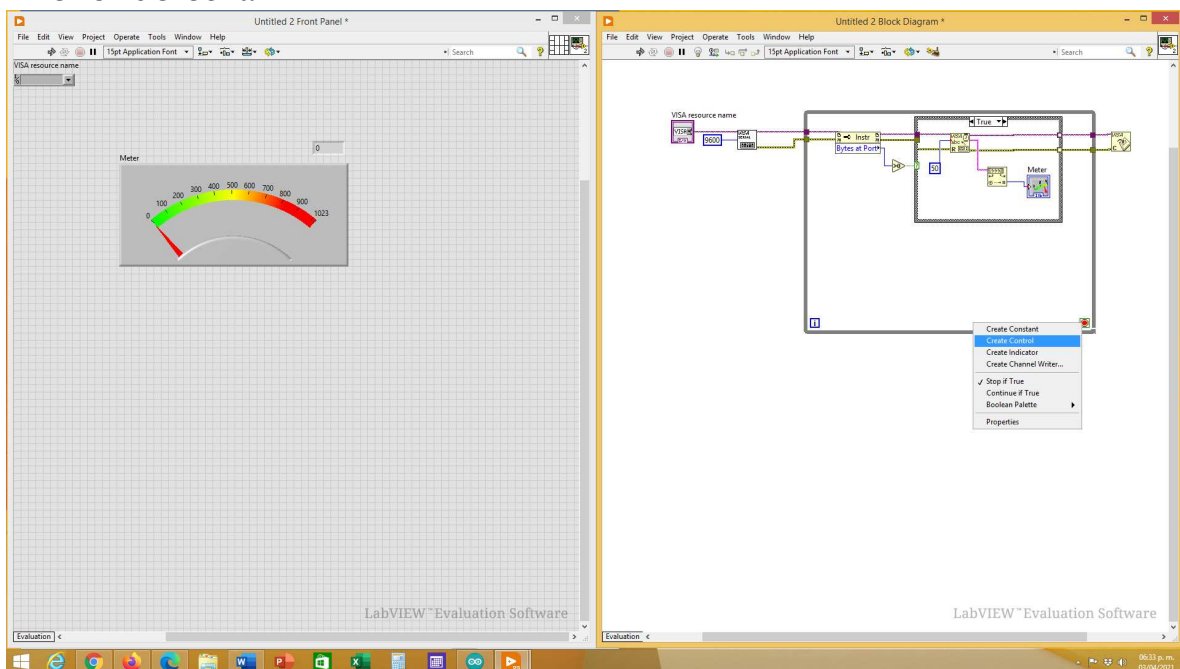
El siguiente paso, sera agregar un convertidor de String a Numero ya que lo que llega por el puerto serial, siempre será texto y necesitamos compararlo



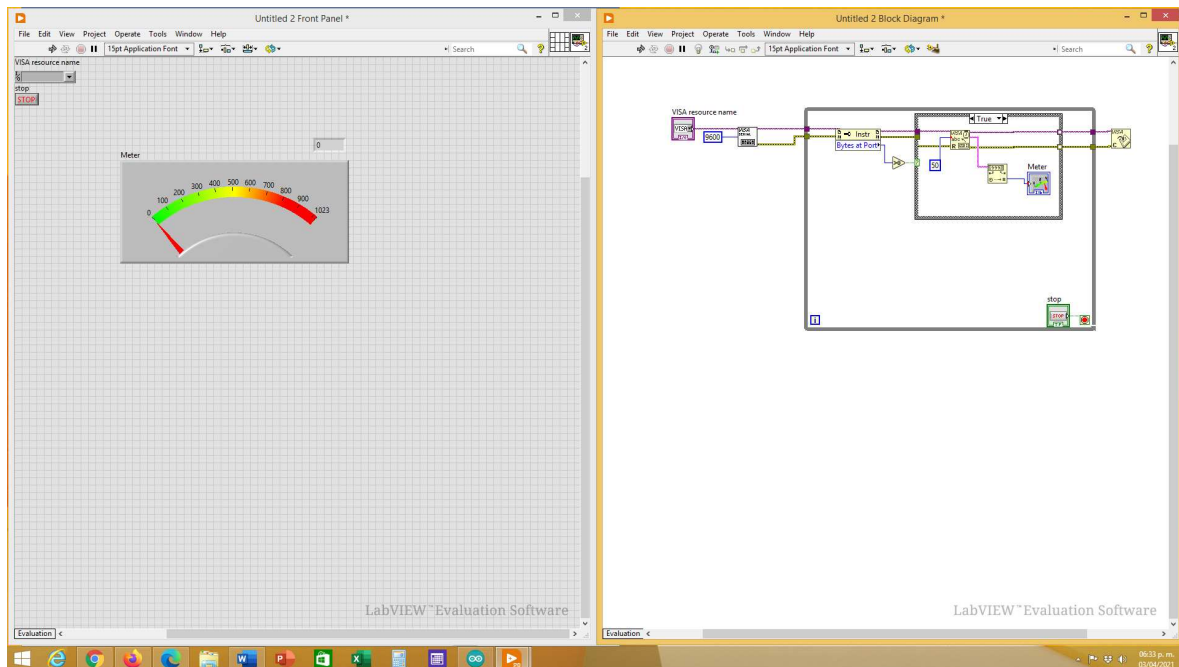
Este bloque ira conectado a la salida de READ y unira ademas al Contol Meter como se muestra en la imagen.



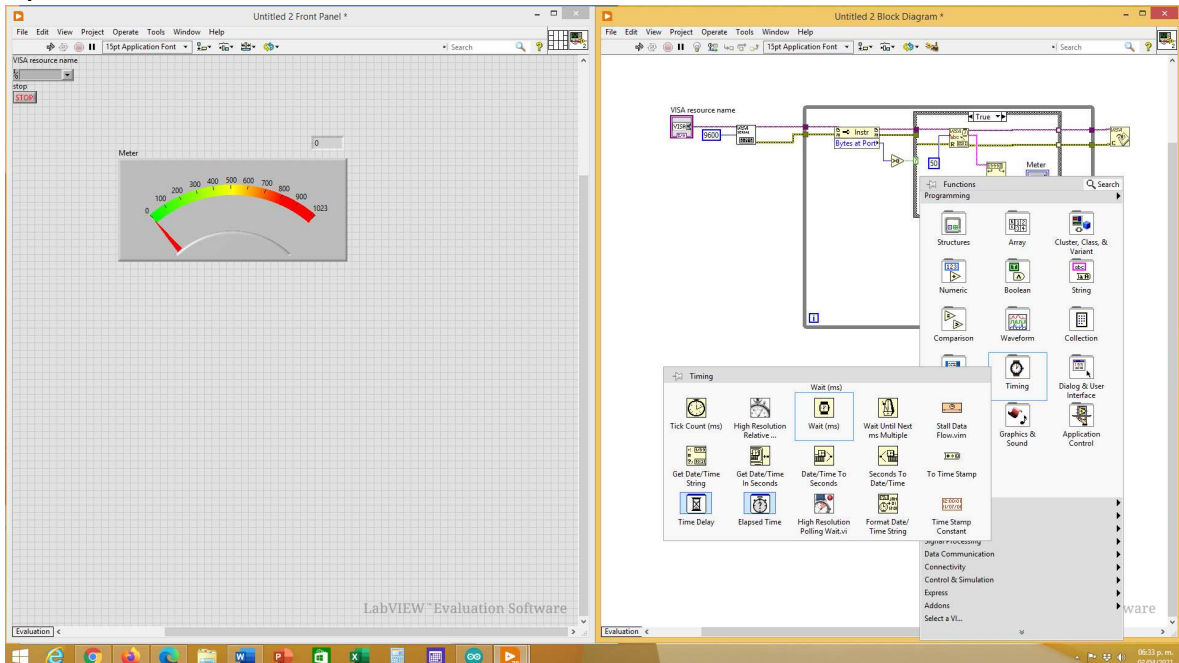
Como pasos finales, agregamos el Control de STOP de la estructura WHILE, dando clic secundario sobre el circulito rojo que aparece en la parte inferior derecha



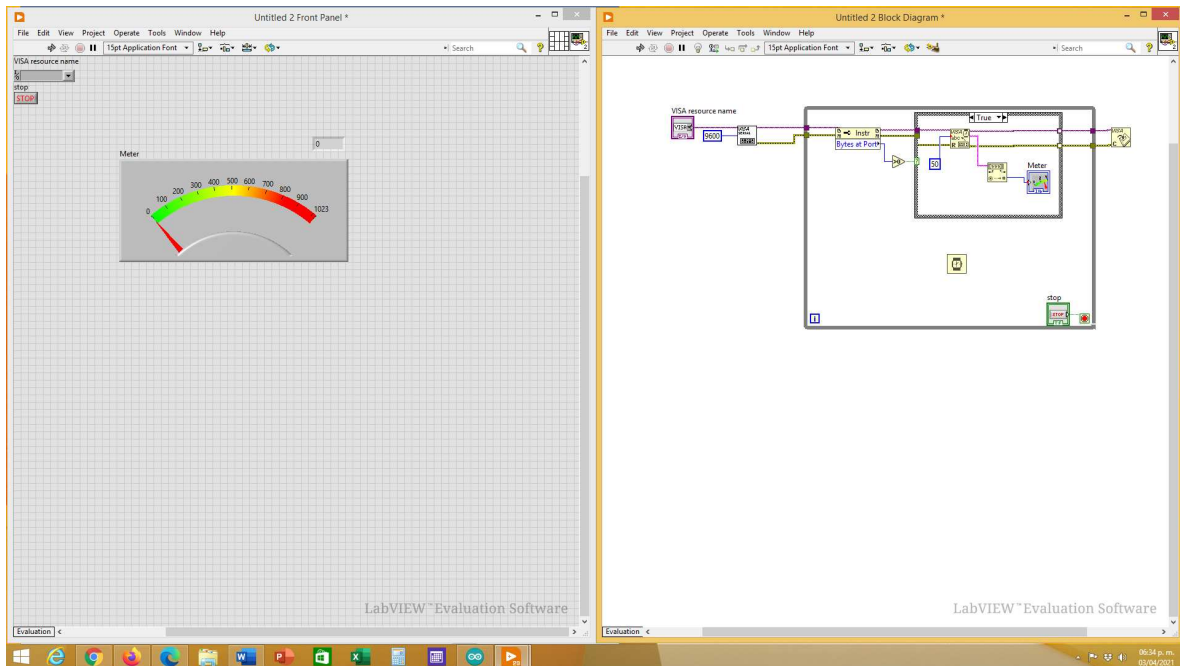
Observemos que se muestra tambien este control en la ventana del Panel Frontal



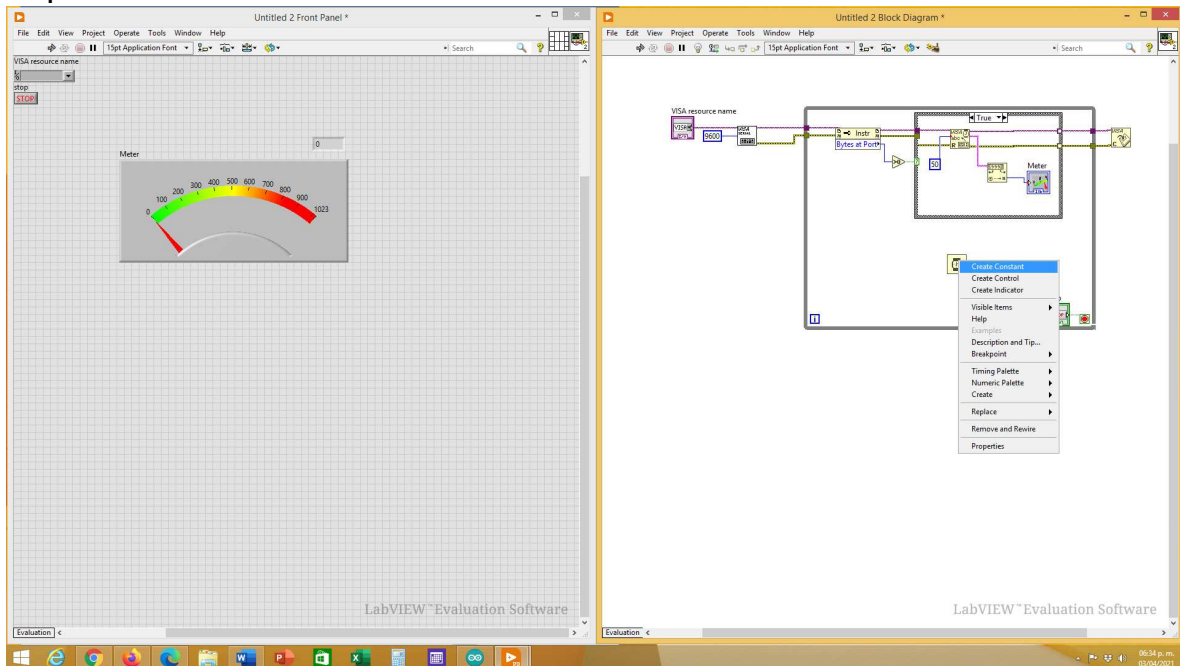
Por ultimo, meteremos un bloque para sincronizar los tiempos de la aplicaci3n de LabVIEW on los de Arduino



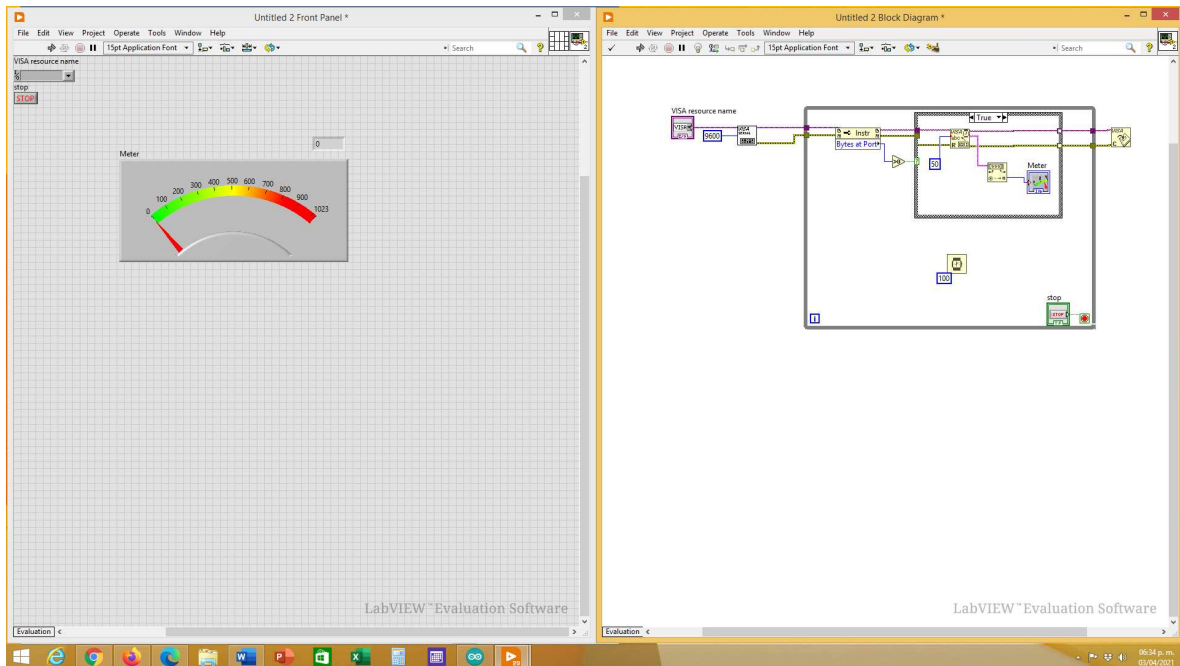
Insertamos entonces un control Wait



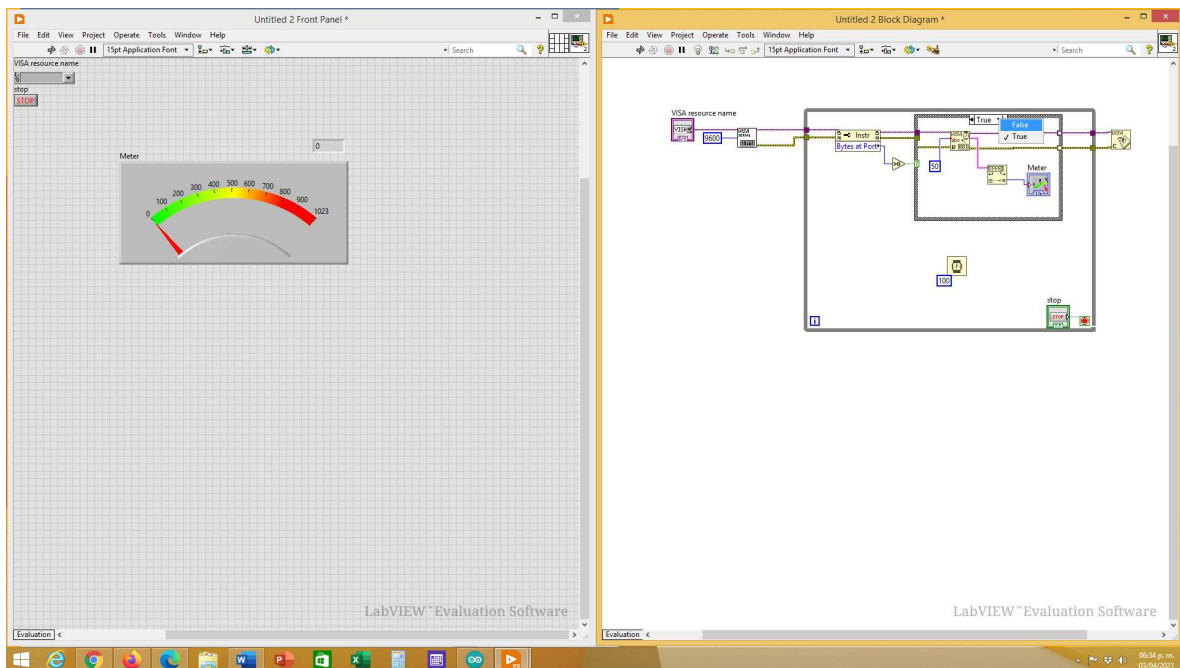
Y agregamos una constante que conectaremos con este reloj. El valor que le pondremos sera de 100ms



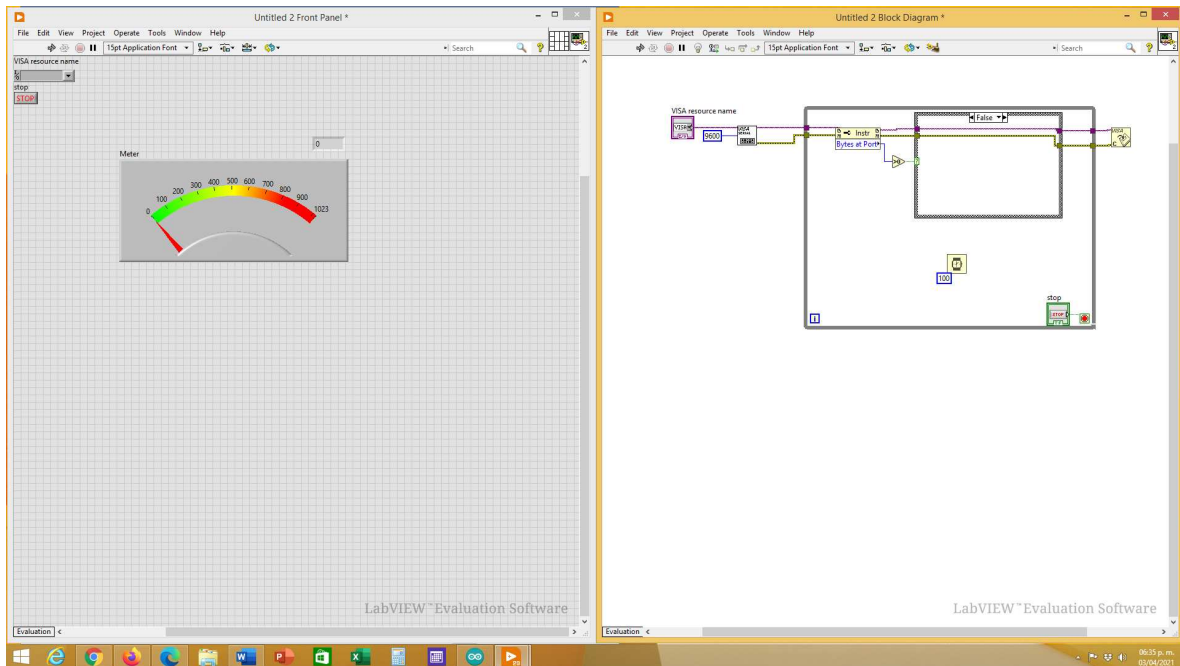
Asi quedarian los bloques



En el CASE donde el valor sea FALSE, vamos a conectar Bytes at Port directamente a CLOSE

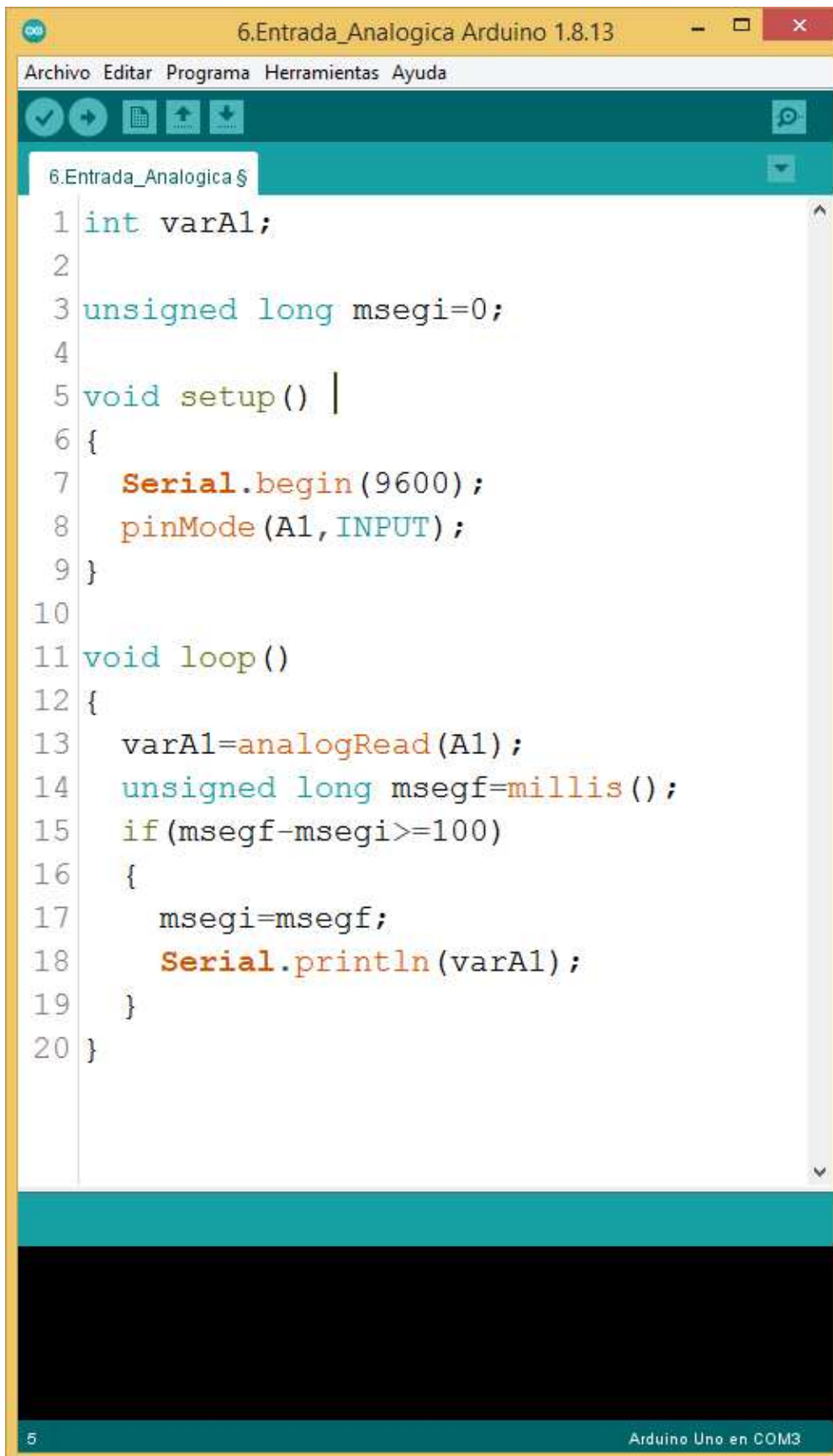


Debe quedar conectado de la siguiente manera



Y por la parte de LabView, eso seria todo.

En Arduino generamos el programa para que lea el valor analogico del potenciómetro (0-1023) y envíe ese valor al puerto Serial. El código quedaria de la siguiente manera:



The image shows the Arduino IDE interface with a file named "6.Entrada_Analogica" open. The code is written in C++ and is designed to read an analog value from pin A1 and print it to the serial monitor every 100 milliseconds. The code includes variable declarations, a setup function to initialize the serial port and pin mode, and a loop function that performs the reading and printing. The IDE's status bar at the bottom indicates that the board is set to "Arduino Uno" and the port is "COM3".

```
6.Entrada_Analogica Arduino 1.8.13
Archivo  Editar  Programa  Herramientas  Ayuda

6.Entrada_Analogica $
1 int varA1;
2
3 unsigned long mseg1=0;
4
5 void setup() {
6 {
7   Serial.begin(9600);
8   pinMode(A1, INPUT);
9 }
10
11 void loop()
12 {
13   varA1=analogRead(A1);
14   unsigned long msegf=millis();
15   if(msegf-mseg1>=100)
16   {
17     mseg1=msegf;
18     Serial.println(varA1);
19   }
20 }
```

5 Arduino Uno en COM3

CODIGO:

```
int varA1;

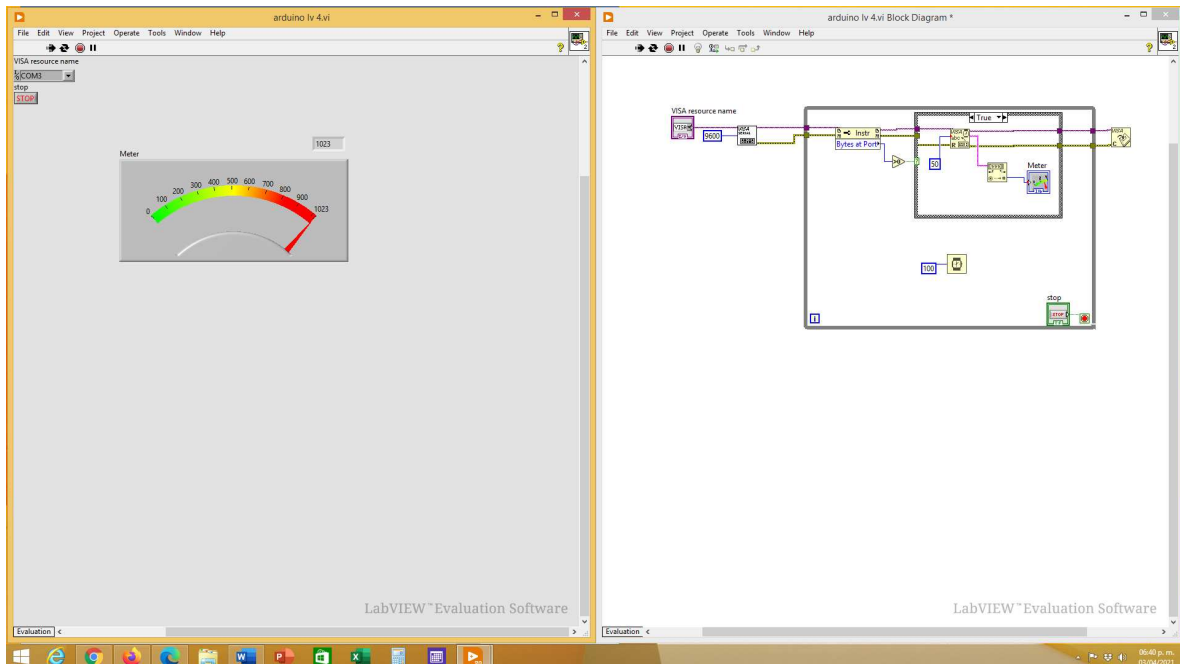
unsigned long msegf=0;

void setup()
{
  Serial.begin(9600);
  pinMode(A1,INPUT);
}

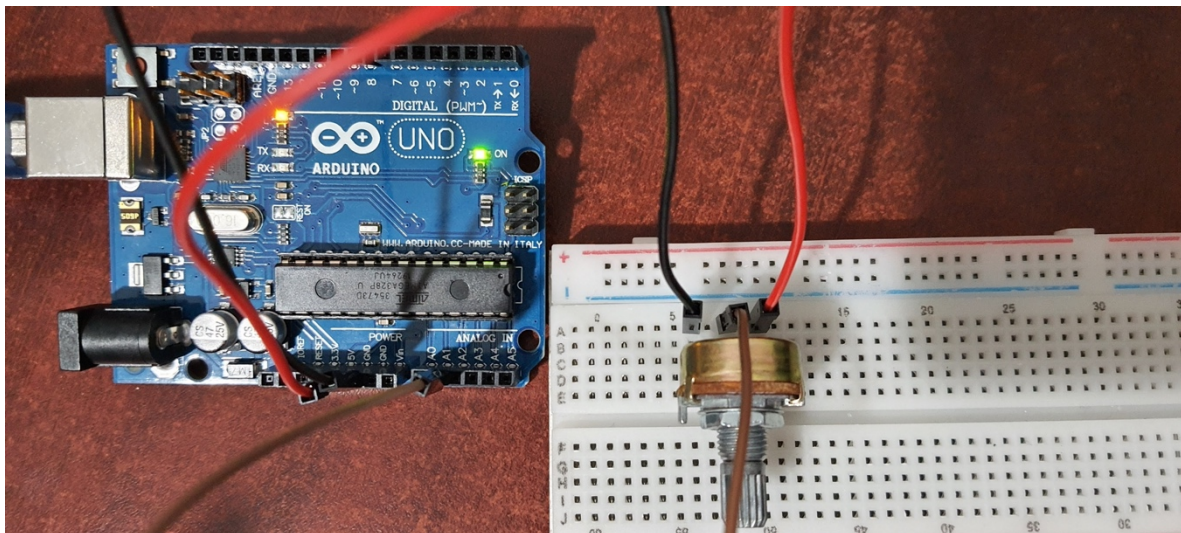
void loop()
{
  varA1=analogRead(A1);
  unsigned long msegf=millis();
  if(msegf-msegf>=100)
  {
    msegf=msegf;
    Serial.println(varA1);
  }
}
```

Una vez cargado el programa al Arduino, ejecutamos la aplicación de labView dando clic en las 2 flechas blancas.

Al mover el potenciómetro, la aguja se desplazara a lo largo de los numeros de 0 a 1023 de manera que siga a la posición de la perilla de dicho control.



Las conexiones del potenciómetro, y los resultados se muestran en las siguientes imágenes.



Graficas de acuerdo a la posición de la perilla del potenciómetro

